

Carnegie Mellon  
Software Engineering Institute

# The 1999 Survey of High Maturity Organizations

Mark C. Paulk  
Dennis Goldenson  
David M. White

*February 2000*

SPECIAL REPORT  
CMU/SEI-2000-SR-002

DTIC QUALITY INSPECTED 2

**DISTRIBUTION STATEMENT A**  
Approved for Public Release  
Distribution Unlimited

20000519 008



Carnegie Mellon University does not discriminate and Carnegie Mellon University is required not to discriminate in admission, employment, or administration of its programs or activities on the basis of race, color, national origin, sex or handicap in violation of Title VI of the Civil Rights Act of 1964, Title IX of the Educational Amendments of 1972 and Section 504 of the Rehabilitation Act of 1973 or other federal, state, or local laws or executive orders.

In addition, Carnegie Mellon University does not discriminate in admission, employment or administration of its programs on the basis of religion, creed, ancestry, belief, age, veteran status, sexual orientation or in violation of federal, state, or local laws or executive orders. However, in the judgment of the Carnegie Mellon Human Relations Commission, the Department of Defense policy of "Don't ask, don't tell, don't pursue" excludes openly gay, lesbian and bisexual students from receiving ROTC scholarships or serving in the military. Nevertheless, all ROTC classes at Carnegie Mellon University are available to all students.

Inquiries concerning application of these statements should be directed to the Provost, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-6684 or the Vice President for Enrollment, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-2056.

Obtain general information about Carnegie Mellon University by calling (412) 268-2000.

This report was prepared for the

SEI Joint Program Office  
HQ ESC/DIB  
5 Eglin Street  
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER



Norton L. Compton, Lt Col., USAF  
SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

Copyright 2000 by Carnegie Mellon University.

**NO WARRANTY**

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

Internal use. Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use. Requests for permission to reproduce this document or prepare derivative works of this document for external and commercial use should be addressed to the SEI Licensing Agent.

This work was created in the performance of Federal Government Contract Number F19628-95-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 52.227-7013.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (<http://www.sei.cmu.edu/publications/pubweb.html>).

---

# Table of Contents

<b>Table of Figures</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>1      Introduction</b>	<b>1</b>
<b>2      Practices of High Maturity Organizations</b>	<b>3</b>
2.1    Management Practices	3
2.2    Engineering Practices	5
2.3    Quantitative Management	6
2.4    People Issues	8
<b>3      Conclusion</b>	<b>11</b>
<b>References/Bibliography</b>	<b>13</b>
<b>Appendix A: List of Maturity Level 4 and 5 Organizations</b>	<b>23</b>
<b>Appendix B: Aggregated Results of the 1999 High Maturity Survey</b>	<b>27</b>
I. Management Practices	28
II. Engineering Practices	34
III. Tools and Technology	43
IV. Quantitative Analysis	51
V. Other Practices	63
VI. People Issues	67
VII. Background Information	78



---

# Table of Figures

Figure 1: Questions VI.1 - Total Number of Employees	67
Figure 2: Questions VI.1.2 - Number of Software Employees	68
Figure 3: Question VI.1.3 - Total Number of Full-Time-Equivalent Software Employees	69
Figure 4: Question VI.2 - Days of Induction Training	70
Figure 5: Question VI.4 - Days of Continuing Education	72
Figure 6: Question VI.6 - Yearly Turnover (Attrition)	73
Figure 7: Question VI.7 - Yearly Growth Rate	74
Figure 8: Question VI.8 - Kinds of Work Space	75
Figure 9: Question VI.9 - Number of People Per Project	76
Figure 10: Question VI.10 - Project Lengths	77
Figure 11: Question VII.1 - Percent of Business Devoted to Software	78
Figure 12: Question VII.12 - Length of Time Doing Process Improvement	79



---

## **Abstract**

Over the last few years the Software Engineering Institute has investigated the high maturity practices of Maturity Level 4 and 5 software organizations via assessments, site visits, workshops, and surveys. This report summarizes the observations from the 1999 survey of high maturity organizations. Areas covered in the survey include management, engineering, tools and technology, quantitative analysis, and people issues. A specific area of interest is statistical process control, which is addressed in some detail in this report, but the observations cover a variety of engineering and management practices, including issues outside the scope of the Capability Maturity Model for Software.



---

# 1 Introduction

During the last several years, the Software Engineering Institute (SEI) has had the privilege of working with a number of high maturity software organizations, as measured by the five-level Capability Maturity Model® for Software (CMM®) [Pault 95], in workshops, conferences, assessments, and site visits. The SEI hosted workshops for Level 4 and 5 organizations in 1996 and 1997. It also participated in various company workshops, which addressed the steps involved in becoming Level 4. SEI staff have also visited a number of high maturity organizations, both informally and during assessments, and had the opportunity to examine their processes in some detail.

When the first profile of maturity levels was published [Kitson 92], no organizations had been assessed as Level 4 and only one organization, IBM's Onboard Shuttle [Billings 94, Fishman 97, Krasner 94, Pault 95, Chapter 6], had been evaluated as Level 5 using the software capability evaluation method.

Six high maturity organizations participated in the 1996 workshop for Level 4 and 5 organizations. The results of that workshop are summarized in Appendix A of "Practices of High Maturity Organizations" [Pault 99a], which primarily reports the 1998 survey of high maturity organizations. The 1997 workshop was held as part of the Software CMM Version 2 effort, and the discussion points are summarized at <URL: <http://www.sei.cmu.edu/cmm/cmm-v2/cmm.v2.html>> in the Software CMM v2 archive.

While workshops and site visits may provide useful insights into industry practices, they do not necessarily provide a good feel for the breadth of deployment of specific techniques across industry. To obtain a broader perspective on these high maturity techniques, an informal survey was distributed in 1998 to Maturity Level 4 and 5 organizations. At the time of the 1998 survey, the SEI assessment database listed 18 Level 4 organizations and 7 Level 5 organizations, which had reported assessment results.<sup>1</sup> A total of 13 organizations responded to the survey.

At the time of the 1999 survey, 61 organizations were known to have been appraised at Maturity Level 4 or 5: 40 at Level 4 and 21 at Level 5. There were 37 respondents to the 1999

---

<sup>®</sup> Capability Maturity Model and CMM are registered in the U.S. Patent and Trademark Office.

<sup>1</sup> A regularly updated maturity profile is available at <URL: <http://www.sei.cmu.edu/sema/profile.html>>.

survey; 18 organizations assessed at Level 4 and 19 at Level 5. The number of high maturity organizations has grown steadily over the last decade, and dramatically in the last two years. As of February 15, 2000, 71 organizations have been appraised at Level 4 or 5, many of which have given permission to be listed in Appendix A of this report.<sup>2</sup>

The detailed aggregate data from the survey is contained in Appendix B. Section 2 of this report summarizes the survey information. While the number of higher maturity organizations is growing rapidly, 37 is still a fairly small sample and should not be over-interpreted. There are also concerns about the consistency and reliability of Level 4 and 5 assessments. This is similar to the situation in 1990, when significant consistency and reliability issues with Level 2 and 3 assessments were reported. This was largely corrected with the publication of Software CMM v1.0 in 1991, which provided a comprehensive description of Levels 2 and 3.

The current release of the Software CMM, Version 1.1, was released in 1993. A conservative stance was taken in defining Maturity Levels 4 and 5 because of the sparsity of Level 4 and 5 organizations. We have learned much about high maturity practices since then, but Levels 4 and 5 are not as clearly articulated in Version 1.1 as we might wish. The CMM Integration work captures much of what was planned for Software CMM v2; drafts of the CMMI model are available on the Web at <URL: <http://www.sei.cmu.edu/cmm/cmms/cmms.integration.html>>. However, the operational model today remains Software CMM v1.1 as released seven years ago. Papers, training, etc., are being deployed to address this problem. For example, SEI courses on high maturity practices and statistical process control for software are now available.

---

<sup>2</sup> Most of the high maturity organizations have provided permission to publish their names, locations, maturity levels, dates of assessment, assessors, and points of contact. That list will be periodically updated and maintained at <URL: <http://www.sei.cmu.edu/cmm/cmm.articles.html#high-mat-orgs>>.

---

## 2 Practices of High Maturity Organizations

Rather than reporting that “73% of high maturity organizations do X,” implying a higher degree of accuracy than is supportable with this small sample, we will use the following terminology in this report:

- “high maturity organizations *typically*...” implies 90% plus of the 4 and 5 organizations have the practice in standardized or common use
- “*most* high maturity organizations ...” implies 60-90%
- “*many* high maturity organizations ...” implies 40-60%
- “*some* high maturity organizations ...” implies more than one

We hope this terminology will provide insight without implying an unjustified rigor.

### 2.1 Management Practices

Although the emphasis of this paper is on good engineering and management processes, it should also be noted that high maturity organizations *typically* have a broader scope of improvement concerns than just CMM process issues. Some high maturity organizations, such as Onboard Shuttle and Boeing Space Transportation Systems, were doing process improvement long before the Software CMM was published. Others, such as Motorola India, were started with one business objective being high process maturity [Paultk 00b]. Responses to inquiries regarding time invested in software process improvement ranged from 2 to 30 years, with a median value of 7 years.

*Most* high maturity organizations have aligned their software process improvement programs with Total Quality Management (TQM) initiatives at the organization or enterprise level.

*Most* high maturity organizations have ISO 9001 certification. Some began their process improvement efforts using ISO 9001, then shifted to the CMM after obtaining certification.

*Most* high maturity organizations use other standards and models to address issues outside the scope of the Software CMM – including ISO 12207 (Software Life Cycle Processes), the People CMM [Curtis 95], and the Software Acquisition CMM [Ferguson 96]. *Most* high maturity organizations build real-time applications and/or embedded systems, and *most* are using a systems engineering model or standard in the process improvement work, such as the Systems Engineering CMM [Bate 95], EIA 731 (Systems Engineering Capability, Part 1: Model), or the INCOSE Systems Engineering Capability Assessment Model.

No particular style of organizational structure dominates high maturity organizations; matrix, functional, product, and customer group structures are the most common.

Software Quality Assurance (SQA) is perhaps the most controversial key process area in the CMM. There are passionately held, opposing opinions on whether there should be an independent SQA organization, or whether the SQA function should be “built into the process” as part of the quality culture to be expected of high maturity organizations.

High maturity organizations *typically* have an independent SQA group and also embed the SQA function in the process. In a typical implementation, process and product assurance are separated, with the “SQA group” focusing on process monitoring, while product assurance is built into peer reviews and/or the configuration management system [Craig 99]. The independent SQA group is usually comparatively small, and it practices sampling rather than providing 100% process and product coverage. The SQA group uses the Level 4 process and product data to identify high-leverage opportunities for auditing. As one reviewer of the 1998 survey commented,

*I was convinced several years ago that SQA would disappear as an organization matures and software quality functions would be embedded into the software engineering functions. I viewed the CMM to have a problem, because SQA as an independent function was always required even as the organization matured (since it was a Level 2 KPA [key process area]). However, I changed my opinion after our experience ... When SQA makes the appropriate transition to process audits in addition to product audits, they can become a valuable team member. We still make mistakes in a high maturity organization, and it is still valuable to have a backup to catch those. Their role in process audits is very valuable as it separates that function from the SEPG (software engineering process group) so that the SEPG does not appear to be the policeman.*

High maturity organizations *typically*

- manage the evolving customer requirements proactively via incremental and/or evolutionary life cycles
- empower project teams to define and use measures in addition to the standard process and product measures, which are not necessarily reported up the management ladder

Most high maturity organizations use

- cost models, such as COCOMO and Price-S
- activity-based costing
- “Top 10 risks” lists. (High maturity organizations *typically* perform systematic risk management, as observed in the 1998 survey.)

- integrated product and process development (IPPD) (a.k.a. concurrent or simultaneous engineering)
- chief architects and chief engineers<sup>3</sup>
- process ownership teams staffed by practitioners, not just SEPG members, managers, and staff. (Worker participation and empowerment builds buy-in and helps overcome barriers to deploying new and improved processes and technologies.)
- project evaluation and review technique (PERT)

*Many* high maturity organizations use

- Delphi methods for estimation [Boehm 81]
- earned value [Thamhain 96]
- ETX, ETVX, or EITVOX<sup>4</sup> for process definition. (Formal process definition methods such as IDEF0 and SADT are used by some organizations, but their formality is frequently perceived as hindering effective communication.)

Process documentation is *typically* easily accessible online, and guidance is readily available when needed. Detailed standards, procedures, and checklists are limited to specific task contexts, such as design inspections or estimating—process descriptions capture the minimum essential information for the user. *Most* high maturity organizations regularly solicit feedback about the usability of process assets.

*Some* high maturity organizations are using the Personal Software Process<sup>SM</sup> (PSP<sup>SM</sup>) and Team Software Process<sup>SM</sup> (TSP<sup>SM</sup>).

## 2.2 Engineering Practices

High maturity organizations *typically* perform peer reviews, as is required of Level 3 organizations. Much of the data used at Level 4 comes from peer reviews. *Most* high maturity organizations perform inspections, the most formal variant of peer reviews, because of their emphasis on collecting data and the associated process rigor, but *many* use both informal peer reviews (e.g., walkthroughs) and inspections.

Formality and data collection/analysis are typical attributes of high maturity inspection processes, and a number of inspection variants have been developed [Fagan 86, Ebenau 93, Freedman 90, Knight 93, Mashayekhi 93]. Gilb's emphasis on inspection sampling [Gilb 93], rather

---

<sup>3</sup> Both IPPD and chief engineers/architects are mechanisms that can be used to break down organizational barriers [Sobek 98].

<sup>4</sup> These methods specify the Entry criteria, Inputs, Tasks, Validation steps, Outputs, and Exit criteria for the process or activity being documented.

<sup>SM</sup> Personal Software Process, PSP, Team Software Process, and TSP are registered service marks of Carnegie Mellon University.

than 100% inspection, to guide process and product decisions is worthy of note, particularly in light of the shift to SQA sampling at the higher maturity levels.

It is a concern that in a few cases the respondent did not know what an inspection or walk-through is. It is also a concern that some comments indicated that peer reviews included managers and/or customers. By definition, “peers” are colleagues at approximately the same hierarchical level in the organization. While designers, coders, and testers may all be considered peers at a professional level, managers who have the authority to hire, fire, promote, and provide raises are not peers, and the inclusion of managers and customers can seriously affect the dynamic of the review. Technical reviews and joint reviews that include managers and customers are desirable in conjunction with peer reviews, but they complement rather than replace the peer review technique.

Most high maturity organizations use

- user interface prototyping
- independent test groups
- code coverage
- domain specific software architectures
- product lines
- systematic reuse (which includes domain specific software architectures and product lines as implementation strategies)

Many of the respondents reported using formal methods. This was a surprising result, given the results of previous surveys and workshops. Follow-up indicates there was ambiguity in how “formal method” was interpreted, and at least some respondents interpreted a “formal method” as any method that was documented. No respondent, in clarifying his or her response, indicated the use of “formal method” in the sense of mathematical rigor, e.g., proof of correctness, that was intended by the question.

*Some* high maturity organizations are using reliability models [Musa 90].

With respect to tools and technology, high maturity organizations *typically* have online repositories of engineering processes and management practices, as well as online defect databases. *Most* high maturity organizations use CASE tools, time sheet automation, databases of commitments with status, and automated data collection tools.

## 2.3 Quantitative Management

Conceptually, Maturity Levels 4 and 5 in the Software CMM are based on statistical process control [Paulk 95, Florac 99], although this was initially stated in terms of operational definitions and comparability in the presence of variation [Humphrey 88]. Level 4 focuses on control—identifying and removing assignable causes of variation in the process, the extraor-

dinary events that prevent the process from performing as intended. Level 5 focuses on improvement—addressing the common causes of variation that are intrinsic to the process. More generally, high maturity organizations appreciate the fundamentals of “statistical thinking”—all work is a series of interconnected processes, all processes are variable, decisions should be based on facts, and a reduction in variation provides improvement opportunities. High maturity organizations are expected to understand the impact of variation on processes and predictability.

The Level 4 key process areas, however, use the term “quantitative management” rather than “statistical control.” The CMM distinguishes between thresholds (desired or expected performance, i.e., “voice of the customer”) at Level 3 and control limits (what the process can do, i.e., “voice of the process”) at Level 4, but the terminology used in the Level 4 practices is “acceptable limits,” “expected mean,” and “expected variation.” Most Level 4 and 5 organizations were initially appraised using a relaxed interpretation of what is meant by “quantitative management” at Level 4.

As time has passed, the expectation for using statistical techniques has grown. Today, high maturity organizations *typically* use control charts and other statistically rigorous methods to control their processes. Control charts used include XmR, XbarR, u and Z charts with no control charting technique predominating. *Most* high maturity organizations are also using other charting methods to understand the “acceptable limits” of variation, such as run charts

*Many* high maturity organizations use

- “cost of quality” (appraisal, prevention, internal failure, and external failure costs) to determine the effectiveness of their process improvement activities
- orthogonal defect classification (ODC) or other defect taxonomies
- prediction intervals
- confidence intervals
- Pareto analyses
- process modeling and simulations

*Some* high maturity organizations use

- quality function deployment (QFD) [Zultner 95]
- designed experiments
- Six Sigma
- analysis of variance such as ANOVA, ANCOVA, and MANOVA
- other multivariate methods

Although controversy remains over what statistical techniques to use when and what business value will be achieved [Ould 96, Carleton 99], the use of rigorous statistical techniques can

now be considered an empirical attribute of high maturity organizations. The initial barrier to using rigorous statistical techniques is “informally stabilizing” the process [Barnard 99, Chatmon 99, Florac 99] to refine operational definitions, make processes more consistent, and minimize variation.

With respect to quality and performance measurement, high maturity organizations *typically* collect data on schedule performance index, requirements stability, and defect density. *Most* high maturity organizations collect data on cost performance index, process stability, rework, customer (or user) satisfaction, and miscellaneous other measures, such as mean-time-to-failure, complexity, and maintainability. *Many* high maturity organizations measure staff morale.

The use of measurement data for evaluating the performance of employees is an ongoing concern for high maturity organizations. Unless a “perfect” measurement system is defined that covers all critical performance parameters objectively, measurement is likely to cause dysfunctional behavior if there is any chance of the data being used against people [Austin 96]. Deming, for example, was a strong advocate of statistical techniques, yet strongly averse to performance evaluations [Deming 86].

## 2.4 People Issues

High maturity organizations recognize the importance of competent people. To quote one participant in the 1996 workshop, “Getting the right person into the right job on the project is still the most important aspect of project success. People are not plug-compatible. The expertise of individuals is critical. Process is an enabler; not a replacement.”

Training in high maturity organizations can go to extremes. Mandatory induction training for new hires ranges from 1 day to 14 weeks (with a median of 6 days and an average of 17, indicating a skewed distribution), plus mandatory continuing education requirements (with a median value of 10 days). High maturity organizations typically require training in technical skills, management skills, and relevant application domains; most also require training in interpersonal skills, team building, and negotiating skills. Training includes internally and externally developed training materials, awareness programs, and workshops. A training program with required training is a key process area for CMM Level 3. Many high maturity organizations (half of those responding) have a formal mentoring program to impart skills and knowledge. Common characteristics of a “formal” mentoring program include

- Mentors are knowledgeable and respected.
- Mentors are trained in how to function effectively in the mentoring relationship.
- The expectations for the mentor and the mentored are explicitly identified.
- The mentoring relationship lasts for an extended period of time, typically about one year.
- Mentor and mentored are physically close together, perhaps sharing an office.

- Mentoring is tracked by management.
- Mentoring skill is part of the performance evaluation criteria for the mentor.
- Causal analysis may lead back to a breakdown in the mentoring process as the root cause of a defect.



---

## 3 Conclusion

What does it mean to be Level 4 or 5? High maturity organizations

- understand why they are doing what they are doing
- know “what to do” when problems are encountered (don’t overreact to special causes – concentrate on finding common causes)
- error-proof their processes to allow for human fallibility
- convert “blame” into “opportunity” (avoid using fear as a motivator)
- balance “empowerment” and “ownership” with “control”
- measure and predict how much further they have to go to achieve their goals

While statistical thinking and an understanding of variation are intrinsic to the definition of Levels 4 and 5, other factors that have been empirically observed—such as capturing product knowledge and addressing the human issues associated with process improvement and change management—are also crucial to continual improvement.

Factors outside an organization’s control are also critical to business success. One of the challenges for any organization is dealing with organizational restructuring—mergers, acquisitions, re-organizations, and rapid growth. Each merger or re-organization can dramatically change the culture of the “original” organization. Onboard Shuttle, for example, was part of IBM when initially evaluated at Level 5, then it became part of Loral, then Lockheed Martin, and it has now become part of United Space Alliance—a dizzying journey over the last decade. Process maturity—and executive recognition of that maturity—can help an organization protect the stability and integrity of its processes during the turbulence of organizational change.



---

## References/Bibliography

**[Austin 96]** Austin, Robert D. *Measuring and Managing Performance in Organizations*. New York, N.Y.: Dorset House Publishing, ISBN: 0-932633-36-6, 1996.

**[Basili 96]** Basili, Victor R. "Applying the Goal/Question/Metric Paradigm in the Experience Factory," Chapter 2 in *Software Quality Assurance and Measurement: A Worldwide Perspective*. Fenton, Norman; Whitty, Robin & Lizuka Yoshinori (editors), ISBN: 1850321744, London, UK: International Thomson Publishing, April 1996.

**[Besselman 95]** Besselman, Joe & Rifkin, Stan. "Exploiting the Synergism Between Product Line Focus and Software Maturity." 95-107 *Proceedings of the 1995 Acquisition Research Symposium*. Washington, D.C.

**[Billings 94]** Billings, C.; Clifton J.; Kolkhorst, B.; Lee, E. & Wingert, W.B. "Journey to a Mature Software Process." *IBM Systems Journal* 33, 1 (1994): 46-61.

**[Boehm 81]** Boehm, B.W. *Software Engineering Economics*. Englewood Cliffs, NJ: Prentice-Hall, 1981.

**[Briand 97]** Briand, L.; El Emam K; Freimut, B. & Laitenberger, O. "Quantitative Evaluation of Capture Recapture Models to Control Software Inspections," 234-244. *Proceedings of the Eighth International Symposium on Software Reliability Engineering*, 1997.

**[Butler 95]** Butler, Kelley L. "The Economic Benefits of Software Process Improvement." *Crosstalk: The Journal of Defense Software Engineering* 8, 7 (July 1995): 14-17.

**[Butler 97]** Kelley Butler. "Process Lessons Learned While Reaching Level 4." *Crosstalk: The Journal of Defense Software Engineering* 10, 5 (May 1997): 4-8.

**[Clark 97]** Clark, Bradford K. "The Effects of Software Process Maturity on Software Development Effort." Ph.D. Dissertation, Computer Science Department, University of Southern California, August 1997.

**[Constantine 95]** Constantine, Larry L. *Constantine on Peopleware*. Englewood Cliffs, N.J.: Yourdon Press Computing Series, 1995.

**[Cosgriff 99a]** Cosgriff, Patrick W. "The Right Things for the Right Reasons: Lessons Learned Achieving CMM Level 5." *Crosstalk: The Journal of Defense Software Engineering* 12, 5 (May 1999): 16-20.

**[Cosgriff 99b]** Cosgriff, Patrick W. "The Journey to CMM Level 5: A Time Line." *Crosstalk: The Journal of Defense Software Engineering* 12, 5 (May 1999): 5-6, 30.

**[Craig 99]** Craig, Rushby "Software Quality Assurance in a CMM Level 5 Organization." *Crosstalk: The Journal of Defense Software Engineering* 12, 5 (May 1999) :11-15.

**[Curtis 95]** Curtis, Bill; Hefley, William E. & Miller, Sally. "People Capability Maturity Model." (CMU/SEI-95-MM-02) Pittsburgh, Pa: Software Engineering Institute, Carnegie Mellon University, September 1995. Available WWW<URL: <http://www.sei.cmu.edu/publications/documents/95.reports/95.mm.002.html>>.

**[Daghfous 91]** Daghfous, Abdelkader & White, George R. "Information and Innovation: A Comprehensive Representation." (Technical Report 91-4). Pittsburgh, Pa.: Department of Industrial Engineering, University of Pittsburgh, 1991.

**[DeMarco 87]** DeMarco, Tom & Lister, Timothy. *Peopleware*, New York, N.Y.: Dorset House, 1987.

**[Deming 86]** Deming, W. Edwards. *Out of the Crisis*. Cambridge, Ma.: MIT Center for Advanced Engineering Study, 1986.

**[Diaz 97]** Diaz, Michael & Sligo, Joseph. "How Software Process Improvement Helped Motorola." *IEEE Software* 14, 5 (September/October 1997): 75-81.

**[Dion 93]** Dion, Raymond. "Process Improvement and the Corporate Balance Sheet." *IEEE Software* 10, 4 (July 1993): 28-35.

**[Dymond 96]** Dymond, Ken. "The Level 4 Software Process from the Assessor's Viewpoint," *Conference of the International Software Consulting Network*. Brighton, U.K., Dec. 1996.

**[Ebenau 93]** Ebenau, Robert G. & Strauss, Susan H. *Software Inspection Process*, New

York, N.Y.: McGraw Hill, 1993.

**[Fagan 86]** Fagan, M.E. "Advances in Software Inspections." *IEEE Transactions on Software Engineering* 12, 7 (July 1986): 744-751. Reprinted in *Software Engineering Project Management*, R.H. Thayer (ed.), *IEEE Computer Society Press*, IEEE Catalog No. EH0263-4, 1988, 416-423.

**[Ferrara 00]** Ferrara, Linda & Timko, Cathy "The Telcordia Technologies Road to Quality." *Cutter IT Journal* 13, 2 (February 2000): 28-34.

**[Fishman 97]** Fishman, Charles. "They Write the Right Stuff." *Fast Company* (December/January 1997): 2-7.

**[Florac 99]** Florac, William A. & Carleton, Anita D. *Measuring the Software Process: Statistical Process Control for Software Process Improvement*, ISBN 0-201-60444-2, Reading, Ma.: Addison-Wesley, 1999.

**[Fowler 97]** Fowler, Kimsey M. Jr. "SEI CMM Level 5: A Practitioner's Perspective." *Crosstalk: The Journal of Defense Software Engineering* 10, 9 (Sept 1997).

**[Freedman 90]** Freedman, Daniel & Weinberg, Gerald M. *Handbook of Walkthroughs, Inspections, and Technical Reviews, Third Edition*, New York, N.Y.: Dorset House, 1990.

**[Gilb 93]** Gilb, Tom; Graham, Dorothy & Finzi, Susannah. *Software Inspection*, Reading, Ma.: Addison-Wesley, 1993.

**[Goldratt 97]** Goldratt, Eliyahu M. *Critical Chain*, Great Barrington, Ma.: North River Press, 1997.

**[Grady 92]** Grady, Robert B. *Practical Software Metrics For Project Management and Process Improvement*, ISBN: 0137203845, Englewood Cliffs, N.J.: Prentice Hall, May 1992.

**[Haley 95]** Haley, T.; Ireland, B.; Wojtaszek, E.; Nash, D. & Dion, R. *Raytheon Electronic Systems Experience in Software Process Improvement* (CMU/SEI-95-TR-017, ADA303319). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, November 1995. Available WWW<URL <http://www.sei.cmu.edu/publications/documents/95.reports/95.tr.017.html>>

**[Haley 96]** Haley, Thomas J. "Raytheon's Experience in Software Process Improvement," *IEEE Software*, 13, 6, (November 1996): 33-41.

**[Hare 95]** Hare, Lynne B.; Hoerl, Roger W.; Hromi, John D. & Snee, Ronald D. "The Role of Statistical Thinking in Management," *ASQC Quality Progress* 28, 2 (February 1995): 53-60.

**[Hauser 88]** Hauser J.R. & Clausing, D. "The House of Quality," *Harvard Business Review* (May-June 1988): 63-73. Reprinted in *IEEE Engineering Management Review* 24, (Spring 1996): 24-32.

**[Hayes 97]** Hayes, Will & Over, James W. *The Personal Software Process (PSP): An Empirical Study of the Impact of PSP on Individual Engineers.* (CMU/SEI-97-TR-001). Software Engineering Institute, Carnegie Mellon University, December 1997. Available WWW<<http://www.sei.cmu.edu/publications/documents/97.reports/97tr001/97tr001abstract.html>>

**[Herbsleb 97]** Herbsleb, James; Zubrow, David; Goldenson, Dennis ; Hayes, Will & Paulk, Mark "Software Quality and the Capability Maturity Model," *Communications of the ACM* 40, 6 (June 1997): 30-40.

**[Humphrey 88]** Humphrey, Watts S. "Characterizing the Software Process," *IEEE Software* 5, 2 (March 1988): 73-79.

**[Humphrey 89]** Humphrey, Watts S. *Managing the Software Process.* ISBN 0-201-18095-2, Reading, Ma. Addison-Wesley, 1989.

**[Humphrey 91]** Humphrey, Watts S.; Snyder, Terry R. & Willis, Ronald R. "Software Process Improvement at Hughes Aircraft." *IEEE Software* 8, 4 (July 1991): 11-23.

**[Humphrey 95]** Humphrey, Watts S. *A Discipline for Software Engineering*. ISBN 0-201-54610-8, Reading, Ma.: Addison-Wesley Publishing Company, 1995.

**[Jose 00]** Jose, Achamma; Anju, N.K. & Pillai, S.K. "Closed Loop Defect Removal Model using SPC," *The SEPG Conference in India 2000*. Bangalore, India, February 22-26, 2000.

**[Kemerer 92]** Kemerer, Chris F. "How the Learning Curve Affects CASE Tool Adoption." *IEEE Software* (May 1992): 23-28.

**[Kitson 92]** Kitson ,David H. & Masters, Steve. *An Analysis of SEI Software Process Assessment Results: 1987-1991* (CMU/SEI-92-TR-24, ADA266996). Pittsburgh, Pa.: The Software Engineering Institute, Carnegie Mellon University, July 1992. Available WWW<URL:  
[http://www.sei.cmu.edu/publications/documents/92\\_reports/92.tr.024.html](http://www.sei.cmu.edu/publications/documents/92_reports/92.tr.024.html)>

**[Knight 93]** Knight, John C. & Myers, E. Ann. "An Improved Inspection Technique." *Communications of the ACM* 36, 11 (November 1993): 51-61.

**[Krasner 94]** Krasner, Herb; Pyles, Jerry & Harvey Wohlwend. *A Case History of the Space Shuttle Onboard Systems Project* (94092551A-TR). SEMATECH, Technology Transfer, October 31, 1994.

**[Langley 95]** Langley, Ann. "Between 'Paralysis by Analysis' and 'Extinction by Instinct.'" *Sloan Management Review* 36, 3 (Spring 1995): 63-76.

**[Langston 96]** Langston, Dale "Framework for Statistical Process Control in a Development Environment," *Level 4 Workshop, Software Productivity Consortium*. March 19, 1996.

**[Masaaki 86]** Masaaki, Imai. *Kaizen: The Key to Japan's Competitive Success*. New York, N.Y.: McGraw-Hill, 1986.

**[Mashayekhi 93]** Mashayekhi, Vahid; Drake, Janet M.; Tsai, Wei-Tek & Riedl, John. "Distributed, Collaborative Software Inspection," *IEEE Software* 10, 5 (September 1993): 66-75.

**[Miller 98]** Miller, Corinne. "Sustaining a Continuous Improvement Culture: From Start-Up Venture to Big Business in a Decentralized Culture," *Proceedings of the 1998 Software Engineering Process Group (SEPG) Conference*. Chicago, Il., March 9-12, 1998.

**[Musa90]** Musa J.D. & Everett, W.E. "Software-Reliability Engineering: Technology for the 1990s," *IEEE Software*. 7, 6 (November 1990): 36-43.

**[Oldham 99]** Oldham, Leon G; Putman, David B.; Peterson, Mark; Rudd, Bruce & Kevin Tjoland, "Benefits Realized from Climbing the CMM Ladder" *Crosstalk: The Journal of Defense Software Engineering* 12, 5 (May 1999) 7-10.

**[Ould 96]** Ould, M.A. "CMM and ISO 9001," *Software Process: Improvement and Practice* 2, 4 (December 1996): 281-289.

**[Paulk 95]** Carnegie Mellon University, Software Engineering Institute (Principal Contributors and Editors: Paulk, Mark C.; Weber, Charles V.; Curtis, Bill & Chrissis, Mary Beth) *The Capability Maturity Model: Guidelines for Improving the Software Process*. ISBN 0-201-54664-7 Reading, Ma.: Addison-Wesley Publishing Company, 1995.

**[Paulk 96]** Paulk, Mark C. "Effective CMM-Based Process Improvement," 226-237 *Proceedings of the 6th International Conference on Software Quality*. Ottawa, Canada, October 28-31, 1996.

**[Paulk 99a]** Paulk, Mark C. "Practices of High Maturity Organizations," *The 11<sup>th</sup> Software Engineering Process Group (SEPG) Conference*. Atlanta, Ga. March 8-11, 1999.

**[Paulk 99b]** Paulk Mark C. & Putman, David "Assessing a Level 5 Organization." *Crosstalk: The Journal of Defense Software Engineering* 12, 5 (May 1999): 21-27.

**[Paulk 99c]** Paulk, Mark C. "Using the Software CMM With Good Judgment." *ASQ Software Quality Professional* 1, 3 (June 19-29, 1990).

**[Paulk 99d]** Paulk, Mark C. "Toward Quantitative Process Management With Exploratory Data Analysis," 35-42. *Proceedings of the Ninth International Conference on Software Quality*. Cambridge, Ma., Oct 4-6, 1999.

**[Paulk 00a]** Paulk Mark C. & Chrissis, Mary Beth. *The November 1999 High Maturity Workshop* (CMU/SEI-2000-SR-003). Pittsburgh, Pa.: The Software Engineering Institute, Carnegie Mellon University, February 2000.

**[Paulk 00b]** Paulk, Mark C. "Indian Software Excellence: Education and Process Pay Off." *Cutter IT Journal* 13, 2, (February 2000): 23-27.

**[Pfleeger 91]** Pfleeger, S.L. "Process Maturity as a Framework for CASE Tool Selection." *Information and Software Technology* (November 1991).

**[Ravishankar 99]** Ravishankar, Sarala; Prashant, Sankaran; Sanjay, Chitnis & Anand, Sampath. "Practices of a CMM Level 5 Organization," *The SEPG Conference in India*, Bangalore, India, February 24-25, 1999.

**[Senge 92]** Senge, Peter. "Building Learning Organizations." *The Journal for Quality and Participation* (March 1992). Reprinted in *IEEE Engineering Management Review* 24, 1, (Spring 1996): 96-104.

**[Shanthakumaran 00]** Shanthakumaran, Prasanth & Daniel, Reuben. "Audit Plan to Process Improvements: An Automated Journey." *The SEPG Conference in India 2000*, Bangalore, India, February 22-26, 2000.

**[Simons 95]** Simons, Robert "Control in an Age of Empowerment." *Harvard Business Review* (March-April 1995): 80-88.

**[Sobek 98]** Sobek, Durward K. II; Liker, Jeffrey K. & Ward, Allen C. "Another Look at How Toyota Integrates Product Development." *Harvard Business Review* (July/August 1998). Reprinted in *IEEE Engineering Management Review* 26, 4 (Winter 1998): 69-78.

**[Thamhain 96]** Thamhain, Hans J. *Engineering Management: Managing Effectively in Technology-Based Organizations*. New York, N.Y.: John Wiley & Sons, 1996.

**[Weinberg 94]** Weinberg, Gerald M. *Quality Software Management Volume 3: Congruent Action*. New York, N.Y.: Dorset House Publishing, 1994.

**[Wheeler 93]** Wheeler, Donald J. *Understanding Variation: The Key to Managing Chaos*. Knoxville, Tn.: SPC Press, 1993.

**[Wheeler 98]** Wheeler Donald J. & Poling, Sheila R. *Building Continual Improvement: A Guide for Business*. Knoxville, Tn.: SPC Press, 1998.

;

**[Wigle 97]** Wigle, Gary B. & Yamamura, George. "Practices of an SEI CMM Level 5 SEPG." *Crosstalk: The Journal of Defense Soft-*

**[Willis 98]** Willis, R.R.; Rova, R.M. et al. *Hughes Aircraft's Widespread Deployment of a Continuously Improving Software Process*. (CMU/SEI-98-TR-006, ADA353168). Pittsburgh, Pa. Software Engineering Institute, Carnegie Mellon University, May 1998. Available WWW <URL: <http://www.sei.cmu.edu/publications/documents/98.reports/98tr006/98tr006abstract.html>>

**[Yamamura 97]** Yamamura, George & Wigle, Gary B. "SEI CMM Level 5: For the Right Reasons" *Crosstalk: The Journal of Defense Software Engineering 10, 8*, (August 1997): 3-6.

**[Zultner 95]** Zultner, Richard E. "Blitz QFD: Better, Faster, and Cheaper Forms of QFD," *American Programmer 8, 10*, (October 1995): 24-36.

**[Zultner 98]** Zultner, Richard E. "Critical Chain – Doing Development Faster With Quality," 26-37. *Joint 1998 Proceedings of the Pacific Northwest Software Quality Conference and the Eighth International Conference on Software Quality*. Portland, Oregon, October 13-14, 1998.



---

# Appendix A: List of Maturity Level 4 and 5 Organizations

The following list of high maturity organizations names most of the Level 4 and 5 organizations that participated in the survey of high maturity organizations. Four of the organizations that responded to the survey have not provided permission to be listed at this time. There were 18 Level 4 and 19 Level 5 organizations that responded to the survey.

The organizations that participated in the survey are noted with a  $\checkmark$  in Column 1 of the table. A more comprehensive list, including points of contact, dates of assessment, and Lead Assessors is maintained at <URL: <http://www.sei.cmu.edu/cmm/cmm.articles.html#high-mat-orgs>>.

As of 15 February 2000, the full list, of which the published list is a subset, includes

- 44 Level 4 organizations
- 27 Level 5 organizations
- 26 non-US high maturity organizations
  - 1 ML4 in Australia
  - 14 ML4 in India
  - 10 ML5 in India
  - 1 ML4 in Israel

Please be aware of the following issues regarding this list.

- The SEI does not certify companies at maturity levels.
- The SEI does not confirm the accuracy of the maturity levels reported by the Lead Assessors or organizations.
- This list of Level 4 and 5 organizations is by no means exhaustive; we know of other high maturity organizations that have chosen not to be listed.
- The SEI did not use information stored within its Process Appraisal Information System to produce this document.
- The organizations listed gave explicit permission to publish this information.
- No information obtained in confidence was used to produce this list.

<b>Survey Participant</b>	<b>High Maturity Organization</b>
✓	BFL Software Limited, Bangalore, <b>India</b>
	The Boeing Company, Aircraft & Missiles & Phantom Works Southern California, Long Beach, CA
	The Boeing Company, Military Aircraft & Missile Systems F/A-18 Mission Computer, St. Louis, MO
	The Boeing Company, Reusable Space Systems and Satellite Programs, Huntington Beach and Seal Beach, CA
✓	The Boeing Company, Space Transportation Systems, Kent, WA [Fowler 97, Wigle 97, Yamamura 97]
✓	CG-Smith Software, Bangalore, <b>India</b>
✓	Citicorp Information Technology Industries Limited (CITIL), Mumbai, <b>India</b>
	Citicorp Overseas Software Limited (COSL), Mumbai, <b>India</b>
	Cognizant Technology Solutions, Chennai, <b>India</b>
	DSQ Software, Chennai, <b>India</b>
✓	Future Software Private Limited, Chennai, <b>India</b>
✓	HCL Perot Systems, Noida and Bangalore, <b>India</b>
	Honeywell International, Avionics Integrated Systems (formerly Allied-Signal, Guidance & Control Systems), Teterboro, NJ
	Hughes Software Systems, Bangalore and Gurgaon, <b>India</b>
✓	IBM Global Services India, Bangalore, <b>India</b>
✓	International Computers India Ltd. (ICIL), Pune, <b>India</b>
✓	Litton Guidance and Control Systems, Woodland Hills, CA
✓	Lockheed Martin Federal Systems, Owego, NY
✓	Lockheed Martin Management & Data Systems, King of Prussia, PA
✓	Lockheed Martin Mission Systems, Gaithersburg, MD
	Lockheed Martin Naval Electronics & Surveillance Systems - Syracuse, Syracuse, NY
	Lockheed Martin Naval Electronics & Surveillance Systems – Eagan, Eagan, MN
✓	Lockheed Martin Naval Electronics & Surveillance Systems – Manassas (formerly Undersea Systems), Manassas, VA

<b>Survey Participant</b>	<b>High Maturity Organization</b>
	Lockheed Martin Naval Electronics & Surveillance Systems – Moorestown, Moorestown, NJ
	Lockheed Martin Space Electronics and Communications Systems – Manassas (formerly Loral Federal Systems), Manassas, VA
✓	Motorola Australia Software Centre, Adelaide, <b>Australia</b>
✓	Motorola, GSM (Global System for Mobile Communications) Systems Division, Network Systems Group, Arlington Heights, IL
✓	Motorola India Electronics Ltd. (MIEL), Bangalore, <b>India</b> [Ravishankar 99]
	NCR Corporation, Teradata Development Division, Massively Parallel Systems, San Diego, CA
✓	Northrop Grumman, Air Combat Systems, Integrated Systems and Aero-nautics Sector, El Segundo, CA
	Northrop Grumman Electronic Sensors and Systems Sector (ESSS), Baltimore, MD
✓	Northrop Grumman, Integrated Systems & Aerostructures, AEW & EW Systems (formerly Surveillance & Battle Management), Bethpage, NY
✓	Oracle Software India Limited, India Development Center, Bangalore, <b>India</b>
✓	Raytheon (formerly Raytheon E-Systems), Garland, TX
✓	Raytheon C3I Fullerton Integrated Systems, Command and Control Sys-tems/Middle East Operations, Fullerton, CA
✓	Raytheon Missile Systems, Software Engineering Center, Tucson, AZ
✓	Satyam Computer Services Ltd., <b>India</b>
	Tata Consultancy Services, HP Centre, Chennai, <b>India</b>
✓	Tata Consultancy Services, SEEPZ, Mumbai, <b>India</b>
	Tata Consultancy Services, Shollinganallur, Chennai, <b>India</b>
	Tata Consultancy Services, US West, Chennai, <b>India</b>
✓	Tata Elxsi Limited, Bangalore, <b>India</b>
✓	Telcordia Technologies, Piscataway, NJ
✓	United Space Alliance, Space Shuttle Onboard Software Project, Houston, TX
✓	US Air Force, Ogden Air Logistics Center, Technology & Industrial Sup-port Directorate, Software Engineering Division, Hill AFB, UT [Paulk 99b]

<b>Survey Participant</b>	<b>High Maturity Organization</b>
√	US Air Force, Oklahoma City Air Logistics Center, Directorate of Aircraft Management, Software Division, Test Software and Industrial Automation Branches (OC-ALC/LAS), Tinker AFB, OK [Butler95, Butler97]
√	US Army, , Communications and Electronics Command (CECOM), SEC, Fire Support Software Engineering, Fort Sill, OK
√	US Navy, Fleet Material Support Office, Mechanicsburg, PA
√	Wipro GE Medical Systems, Bangalore, <b>India</b>
	Wipro Technologies, Enterprise Solutions Division, Bangalore, <b>India</b>
√	Wipro Technologies, Global R & D (formerly Technology Solutions), Bangalore, <b>India</b>

---

## Appendix B: Aggregated Results of the 1999 High Maturity Survey

The following two sets of definitions were used to describe how often the practices are used in the organization. The first set focuses on the approach—how the process is defined.

<i>Standardized</i> ( <i>Std</i> )	The practice is institutionalized as part of the organization's standard software process. The practice is expected to be followed whenever an opportunity for its effective use arises. Instances where it is not followed are rare exceptions, e.g., in legacy systems or when customer requirements dictate the use of other practices.
<i>Common use</i> ( <i>Comm</i> )	The practice is followed frequently, or even in almost all instances, when it is appropriate. But it cannot be considered as an institutionalized, standard operating practice in the organization.
<i>Not typically used</i> ( <i>Not</i> )	The practice is not typically used throughout the organization. It may be used infrequently, perhaps under special circumstances, or on an ad hoc basis.
<i>Don't know</i> ( <i>D/K</i> )	The respondent was not familiar with the practice or aware of its use in the organization.
<i>Not applicable</i> ( <i>N/A</i> )	The practice has been judged as being not applicable for the organization.

The second set of definitions focuses on deployment – how widely the process is followed.

<i>Ently</i>	Almost entirely
<i>Large</i>	Large extent
<i>Some</i>	Some extent
<i>Limit</i>	Limited extent
<i>Hrdly</i>	Hardly at all

Most survey questions had 37 responses. N indicates the number of responses to a question, when it differs from 37.

## I. Management Practices

1. First of all, how are the following management practices used in your software organization?

		Std	Comm	Not	D/K	N/A
1.1	Cost models (e.g., COCOMO, COCOMO II, Price-S, SLIM, or SPR)	15	10	10	1	1
1.2	Delphi methods for estimation	9	11	15	0	2
1.3	Earned value	17	5	14	0	1
1.4	Activity based costing	19	8	7	1	2
1.5	"Top 10" risks lists	25	7	4	1	0
1.6	Personal Software Process <sup>SM</sup> (PSP <sup>SM</sup> ) and/or Team Software Process <sup>SM</sup> (TSP <sup>SM</sup> )	0	1	31	5	0
1.7	Integrated product and process development (IPPD) (a.k.a., concurrent or simultaneous engineering)	16	11	5	2	3
1.8	Chief architect / chief engineer	16	10	10	0	1
1.9	Independent SQA group	35	0	2	0	0
1.10	SQA function embedded in process (e.g., a role in the peer review method, via buddy system, or as Software Configuration Management entry criteria for baselining)	34	2	1	0	0
1.11	Proactive management of evolving customer requirements (e.g., incremental or evolutionary life cycles)	22	14	1	0	0
1.12	Process ownership teams (i.e., staffed by practitioners, not just SEPG and management)	23	9	3	1	1
1.13	Project evaluation and review techniques (PERT) (e.g., via Microsoft Project)	22	10	4	0	1
1.14	Project teams empowered to define and use non-reported meas-	19	15	3	0	0

		Std	Comm	Not	D/K	N/A
	ures (i.e., to increase their process insight and understanding -- in addition to standard reported measures)					
1.15	IDEF0 or SADT for process definition	3	3	18	7	6
1.16	ETX, ETVX, or EITVOX for process definition	17	4	9	3	4

**2. What, if any, management practices that may prove to be important for performance excellence are currently being piloted in your organization?**

Enterprise Project Management: External and Internal Customer Satisfaction programs.  
Automated audits using event analysis tools.

Six Sigma Tool Box: QFD - Quality Function Deployment; DOE - Design of Experiments; FMEA - Failure Mode Analysis; Statistical Tools Like ANOVA, T TESTs, F-Tests, - e.g., Minitab etc.; Process Mapping; CTQ - Critical to Quality Requirements Flowdown Technique

CMMI and TSP. There is also a new ISO standard 12207? that we may start reviewing in the near future

Use of an organization board to consider candidates from within the company and establish some of them as qualified and certified Program Managers and Chief Architects. The selection is based on education and experience level and is intended to ensure a fresh flow of recognized candidates for these positions.

6 sigma blackbelt classes on going on with the idea that these blackbelts will act as interventions for problematic process areas. Knowledge management tools (HyperKnowledge) are being piloted as a mechanism to define processes and capture rationale for selection of process elements.

Software process modeling and simulation.

Implementation of an automated metrics collection, distribution and analysis capability using DDS's MetricsCenter tool.

Activity based costing.

Raytheon 6 Sigma program (overall continuous measurable improvement program). Raytheon Systems Company Integrated product development system.

Restructuring and refining of Integrated Product Teams for more effective implementation. Includes team launches, charters, training, and organization changes.

Numeric Management with the help of Satyam's Vision Compass tool. Automated process assessment across the company for CMM Level 5. Earned Value.

Tools and Technology for the productivity enhancement is being piloted in the organization.

Common causes of defects across the org are being studied.

SPC for Project Management

Strength: Problems are blamed on the processes and never on the people

The following are institutionalized:

- PPDP : Performance Planning and Development Program
- AOE : Award of Excellence presented on NIIT Annual Day (Nomination)
- MQDC : Managing Directors Quality Club (Nomination)on Annual Day
- PC : Presidents Club (selected by Peer voting) on Annual Day
- IEF : Individual Effectiveness Feedback (270 degrees feedback system)
- PQI : Personal Quality Initiative

We have implemented the Project Office concept which has now gone beyond the initial pilot stage and stabilized. Project Office provides central information control for all project activities. This helps the management in making any decisions.

User satisfaction survey on process management activities to improve internalization of processes. Quantification of Process Improvement.

Results Management System for performance appraisal of individuals driven by Role Description that defines scope and skills required of the roles in the company.

Start up reviews and non advocacy reviews have individuals from other organizations with no vested interest in the program review the contents, scope and plans for the program. Recommendations are made to risk identification, staffing, domain expertise, and process issues.

Enterprise wide integrated project planning and activity measurement system

Raytheon 6 sigma process: goal to reduce waste; ensure all activities value added. SE-CMM.

Balanced scorecard approach for performance excellence

Following are standardized practices in the organization (focus for review on process & quality related activities)

- Detailed checklist driven project Management reviews
- Monthly division wide Quality Council meetings
- Quarterly Quality review meetings
- Six monthly organizational level Management Review Meetings
- Monthly Management council meetings (more focus on business)

Apart from this we have periodic Employee perception survey and open house meetings where Senior management meets all employees.

High potential employees are being used as Malcolm Baldridge (High Performance Business Systems) assessments and corrective action determination.

Piloted various cost models to determine best fit for our domain and development process.

**3. Has your organization piloted or otherwise considered these or other management practices but rejected them for common or standardized use? (Please describe any such practices here, along with your reasons for rejecting them)**

Function Point Estimation: Errors in Practice; Did not Work.

Note: Question 4.5 is difficult "Never" versus "Now". Over the years we have greatly reduced the need to have hardcopy processes but some engineers still make printouts of the documents, and the hardcopies are often provided to new employees as a part of their training.

ABC has been considered, but apparently the financial systems of the organization aren't setup to accommodate it.

IDEF0, SADT: We found they were difficult to communicate with (lower maturity) customers. Piloted many specific cost models and selected SEER, COCOMO as most realistic for our type of business.

Schedule Management methodology and associated cost and schedule management package (tool). Problems with training of people and bringing new people up to speed, work station capacity and implementation difficulties. In general, were not impressed with results.

Distributed SEPG - an SEPG formed with part time members from projects

We try out new practices after initial evaluation and commitment. After that we may improve them further. We have rejected some of the models found successful in other organizations, because they were not culturally suitable or they were very "expensive"

MBO was tried and rejected after we found they did not provide comprehensive practical framework.

We reviewed PSP, but the engineers did not like it.

Considered PSP, but have not pursued considering the effort/investment required and uncertainty of impact on the bottom line.

Earlier we had organization wide monthly Quality review meetings which were made division wide to give more focus at the division wide on quality related activities. Quality review meetings are now held once in a quarter at Org level.

Answer to this question is NO. Other comments regarding above: We practice Team Software Process in our own terms and have for several years, we call it Feature Team process. Also, we have an independent SQA at the same time we have also integrated SQA functions into the process... both co-exist.

**4. Following are a series of statements that describe the use of process documentation in software organizations. How well do they describe your organization? (Please select one for each.)**

		Ently	Large	Some	Limit	Hrdly
4.1	Our process documentation is easily accessible online (e.g., via the web or other networked services).	33	2	2	0	0
4.2	Process guidance is made readily available when needed.	30	6	1	0	0
4.3	Detailed standards, procedures, or checklists are limited to specific task contexts (e.g., design inspections, estimation procedures, baselining).	25	10	1	0	1
4.4	Since improving our process maturity, we now depend less on hard copy process documentation (e.g., manuals).	19	11	4	3	0

		<b>Ently</b>	<b>Large</b>	<b>Some</b>	<b>Limit</b>	<b>Hrdly</b>
4.5	We never had much hard copy process documentation. (N=34)	7	8	6	5	8
4.6	Hard copy process documentation is updated regularly (N=36).	16	3	5	4	8
4.7	Our SEPG regularly solicits feedback about the usability of our process assets.	17	14	5	1	0

## II. Engineering Practices

### 1. How are the following engineering practices used in your software organization?

		Std	Comm	Not	D/K	N/A
1.1	User interface prototyping	8	23	5	1	0
1.2	Independent test groups (i.e., independent of the developers of the software system)	26	5	6	0	0
1.3	Code coverage	14	17	5	0	1
1.4	Reliability models ( $N=36$ )	5	8	20	1	2
1.5	Frequent (e.g., daily) build & smoke tests ( $N=36$ )	7	11	11	5	2
1.6	Peer reviews (e.g., walkthroughs, Fagan-style inspections, Gilb-style inspections, active design reviews, or variants thereof) (Please describe briefly here.)	36	1	0	0	0
1.7	Formal methods	19	6	11	0	1
1.8	Domain specific software architectures	14	17	4	1	1
1.9	Product lines ( $N=36$ )	14	10	6	6	0
1.10	Other systematic reuse (i.e., characterized by an organizational strategy for reuse) (Please describe briefly here.)	9	14	14	0	0

#### Comments on question II.1.6 – Peer reviews

Technical doc review and code walkthrough standardized

Fagan and a non-technical review method used.

Variant of Fagan Style Inspection called HIGH IMPACT INSPECTION OR HII. Includes, not only peers but ALL STAKE HOLDERS including management & customers, e.g. Even Design Documents are explained to customer & reviewed.

Fagan-Style is the most commonly used method

Life cycle milestone formal peer reviews. We call them Quality Point Reviews.

Fagan style inspections

Use desk checks for small, non-complex changes. Use modified "Rifkin" method (which is a Fagan-style) for meetings.

Our peer reviews are called Engineering Technical Reviews (ETRs). Common process for all peer reviews with checklist for each. Individuals trained in the process and their roles and responsibilities

Walkthrus, checklist based reviews based on documented procedures for different types of artifacts

Structured walkthroughs

All of the above.

Fagan style inspections for specifications, design and critical code modules

Fagan-style inspections and Walkthroughs

A variation of Fagan style inspection, with a lot of rigor on the review process itself but less rigor on the prework for the peer review Fagan inspections

Joint reviews. Independent reviews. Fagan-style inspection

Variant of Fagan style inspection. Inspectors are given roles and the file name to be inspected. Performed at three levels: requirements, design, code. All issues are documented on action item form by inspectors. Time to inspect is generally four to five days, but is governed by size of function being inspected. Meeting time is limited to clarification of action items if not understood by author. Always kept to less than 30 minutes.

Fagan style Inspection is standardized for certain work products

Fagan

Fagan-style inspections

Variant of Fagan-style inspections - formal defined roles, trained participants and defect identification and collection.

Fagan style inspections of design & code required; recommended for requirements, test plans & procedures

We do extensive peer reviews of all deliverables. Our customers, (development partners) are members of our peer review team. Reviews are quantitatively controlled and data is used for process management, causal analysis & defect prevention. Informal PSP happens as a result of the visibility of the data from peer reviews.

Structured walkthroughs

Fagan-Style inspections. Roles defined in reviews. Metrics data collected during reviews, checked against norms and analyzed.

Mostly peer walkthroughs and technical stakeholder reviews

All work products are subject to extensive peer reviews including 100% of code

Fagan inspections.

Walkthroughs and inspections performed on all software work products.

Primary mechanism is code inspection / unit test inspection, emphasizing two major points: To prove the fix/enhancement did what it was supposed to, and to prove it did not cause a 'ripple effect.' We also peer review designs when they are significant enough.

Formal Inspections done on front end, formal walkthroughs thereafter.

#### **Comments on question II.1.7 – Formal methods**

In my mind formal methods implies a method employing formal proofs. That is not what we do. However it was the opinion of the group assigned to answer the questions that a formal method could be any method which is formally documented and used by a cross section of organizations across the industry such as UML, or in our case ADARTS. We use ADARTS with limited tailoring and follow the prescribed entry and exit criteria for all activities. In that sense our method is formal. We do not use formal proofs or associated techniques.

Our organization does not use formal methods in the sense of mathematically provable correct design. We use formal inspections and a defined (formal) method for estimating effort. There are other parts of our software development process that are formal (documented) as opposed to informal.

If it was peer reviews, we do Fagan inspections, have developed an in-house tool to conduct them, record findings and track actions. If it was QPM/SQM, we have developed mathemati-

cal models for projecting latent defects and so on... We years ago experimented with mathematical proofs of SW correctness, but it never got past the theory stage.

We interpreted formal methods as use of the following: XmR charts, Histograms, Pareto Diagrams, Bar Charts, Scatter Diagrams, Orthogonal Defect Classification.

We are using "Executable Formal Descriptions."

We do carry out Fagan type inspections for reviews. These are not mandatory in all cases. Also, we are using SEI method for Interim Assessments at regular intervals - 4 in the last 15 months.

#### **Comments on question II.1.10 – Other systematic reuse**

We do reuse in a group of projects in a particular application area

The reuse strategy is commonly used during development and our QPM/SQM/DP meetings consider reuse

DII COE

We have three business areas, two of which are product line oriented and the other uses functional requirements.

Process defines activities at project startup to access the process assets library, and in-project and wind up reviews to identify process assets for storage in library. Library is organized subject wise.

Had a multiyear program to analyze software reuse strategies that resulted in an update to our common software process. However, we realized that scope of effective, systematic reuse process would be enterprise-wide, and require organizational changes that have been overcome by events.

Have a standard, documented process and a reuse library.

Organization is working towards this.

Part of the project plan itself

Reuse Working Group exists at an organization level which maintains the reuse repository, and facilitates projects in defining reuse strategy. The project plan template has a section on reuse planning for the project also.

We have a reuse process, which describes our approach at a high level

A formal re-use strategy exists. Usage of Intranet helps the process. Data collection on re-use patterns helps a dynamic correction.

a) Assets library populated with reusable project documentation and its use facilitated and monitored. b) IBM intellectual capital accessible for project learning and processes.

No systematic reuse. Reuse by opportunity is common place.

Software Reuse Repository on a Sun System

Reusable components established and designed for use across application tools in OO environment

COTS/GOTS First initiative forces management to look at reuse when proposing, re-planning or researching a new product.

We are focussing on the twin lifecycle model for reuse. This needs business maturity, alignment in business, technology & implementation strategies for maximum leverage. Domain strategies based on all these factors are chosen as appropriate.

We have query based Historic Projects database to drive Defect prevention and process optimization based on earlier projects learnings and best practices. Apart from this we also have project level and group level homepages which help in promoting reuse

Components developed as a strategy in respective Technology groups

Reuse at a high level across product lines and reuse at a very low level for basic functions.

We are actually putting a lot of attention here as a major initiative.

We only have one domain and one product line - all of which is unique to IUS. Not exactly the kind of environment that allows reuse.

**2. What, if any, engineering practices that may prove to be important for performance excellence are currently being piloted in your organization? (Please describe them here)**

OO/UML/Architectures.

SCORECARDS. We have designed MS EXCEL based score cards to measure every quality goal to the last function level in quantitative way, which is flown down from CTQ or Critical

To Quality Requirements. This is the key driver for EVERY engineer to check if he has met his personal Quality Goals, which in turn are aggregated at Project level. We have 4 types of such score cards: PROCESS (Escape ratio etc.), INTERNAL PARTS (software components developed anew), DESIGN PROGRESS (points to FMEA & Test Completeness/Coverage of Code).

CMMI and TSP

Use of UML

Reliability Models

Innovative software safety program. New reliability modeling and analysis approaches. DII COE

SE Process Self-Assessment methodology; Trade study methodology with supporting tool set; Operation description methodology with supporting tool set; CRADLE - CASE SE development tool; EIA/IS 731 Assessment Methodology; Automated counting of Function Points

Review efficiency to be improved from the current level of performance.

Strength: Verification and validation at every phase

ASDM : Application S/W Development Methodology, an in-house methodology for rapid application development, that is closely coupled with an in-house tool called SETS, which automates application generation from specifications (available for environments like Power-builder, Visual Basic, Java and ASP) Testing Techniques for improving the quality of testing.

SE-CMM (EIA 731) and CMMI. Also a "hardware CMM" which uses CMM-like criteria on hardware development.

To take care of a continuous need for changing requirements, we have implemented on pilot basis, an Iterative Development Method. This uses concepts like, very short release cycle, overlapping phases and late finalization of requirements.

Reliability models are being piloted. IBM WSSDM Tool with different life cycle models is built. Different OO Models. FFP for Estimation

Code Reuse

Piloting Rational SODA software documentation.

Architecture Based Design - when many defects were found during test that affected design, a more rigorous design process was identified. In-Process Quality - because inspections were so well defined, when we kick off a team at the beginning of a phase

An enterprise wide defect logging, tracking, and analysis tool is planned.

Object-oriented design with incremental lifecycle is moving from pilots to common usage.  
Concurrent engineering

We have a performance scorecard which looks at all aspects critical to the success of the business. Software goals like reuse, 'non-handwritten code' cycletime, apart from standard quality goals are disseminated throughout the organization as a result of the scorecard approach. Domain strategies push the technology maturity of each domain. This provides extensive focus on the use of technology, automation, architectures and component based development approaches. Product quality management is focussed and world-class technologies are used for managing many of the quality attributes.

We are using Six Steps to Six Sigma (SSSS) methodology to bring in performance improvement in certain software engineering practices. Typically we take this as time bound Six Sigma project to be run by a set of trained practitioners ( project team members) and facilitated by a trained Black Belt in this Six Sigma methodology. Currently we completed one project on Technical review Effectiveness. Other two projects which are being piloted now are 1. Reduction of Rework effort and 2. Reduction of Post release defects.

None. Our peer review process (we call it the Design Review Board - or DRB) was created back in 1985-86 time frame, and has proved to be the single best practice that improved quality and reduced defects. The DRB is basically 4-5 of our most senior technical experts who you must convince that your work was correct. We basically reuse this mechanism for a wide variety of tasks, not just software development. Anything we release that we think is important gets "DRB'd".

- 3. Has your organization piloted or otherwise considered these or other engineering practices but rejected them for common or standardized use? (Please briefly describe any such practices here, along with your reasons for rejecting them)**

Fagan-style inspections - Overly formalized, didn't yield the expected benefits.

Formal specification/design language use - Tool was too expensive, ADA use went out of style; Clean room methods - reliability of Mean Time Between Failure (MTBF) measurements on software and requirements instability

PSP: 3 staff members of a project team piloted PSP in their project (by enrolling for the distance learning program with SEI). We found the results beneficial, but could not design an effective way to institutionalize the program.

No. Planning on piloting this coming year.

Any engineering practice goes through a formal method described in our Standards and Procedures manual described under Process Change Management. A quantitative evaluation at the end of pilot results into a decision to continue or reject the new idea. Since we spend some effort in evaluating the idea before the implementation, so far we have not rejected any idea after the pilot.

Certification of internal faculties for training and orientation was considered and rejected. We found this will reduce the enthusiasm of the internal faculties.

Formal methods was rejected after piloting due to heavy dependence on external requirements generation and requirements specification/notation not being conducive to widespread deployment of the technique

Internal attempts to implement workflow automation for the technical development processes were rejected, since the development processes should be able to accommodate ambiguities, changes and imprecise definitions to some extent and cannot be 100% automated

We had come with one process called FTMP (Fast Time to Market Process) which was run as trial process and was not standardized, as it did not meet the expected objectives.

Before the DRB we did straight Design reviews - but because of the technical complexity of the design, most review participants didn't understand the system well enough to give meaningful feedback, or didn't understand it at all and glazed over. That's why the DRB has technical experts on it - because they can understand.

Question for question 4 below: I'm not sure I understand the definitions. I will assume that the walkthroughs are the same as peer reviews and inspections are independent reviews.

#### 4. What kinds of work products are subject to inspections or walkthroughs in your organization?

	Work Prod- ucts	Peer Review Technique	Std	Comm	Not	D/K	N/A
4.1	Requirements	Inspections (N=35)	27	1	6	1	0
		Walkthroughs (N=25)	13	4	8	0	0
4.2	Design	Inspections (N=35)	27	2	5	1	0

	<b>Work Prod- ucts</b>	<b>Peer Review Technique</b>	<b>Std</b>	<b>Comm</b>	<b>Not</b>	<b>D/K</b>	<b>N/A</b>
		Walkthroughs ( <i>N</i> =25)	10	7	7	1	0
4.3	Code	Inspections ( <i>N</i> =34)	26	3	4	1	0
		Walkthroughs ( <i>N</i> =29)	19	7	3	0	0
4.4	Test (e.g., cases, plans, or procedures)	Inspections ( <i>N</i> =34)	23	2	8	1	0
		Walkthroughs ( <i>N</i> =27)	13	6	8	0	0

### III. Tools and Technology

#### 1. How are the following tools and technologies used in your software organization?

		Std	Comm	Not	D/K	N/A
1.1	CASE tools (e.g., upper or lower) (Please describe briefly here) (N=35)	13	12	8	2	0
1.2	Online repository(is) of engineering processes and management practices	35	2	0	0	0
1.3	Time sheet automation to collect effort data in useful categories	32	1	4	0	0
1.4	Database(s) of inter-group and intra-group commitments and their status	12	11	11	3	0
1.5	Automated data collection tools (e.g., on-line forms with "tickler" reminders, time-stamped activity logs, static or dynamic analyses of call graphs or run-time behavior)	10	10	16	1	0
1.6	Online defect database(s) (e.g., injection or removal)	27	7	3	0	0
1.7	Other process automation (Please describe briefly here) (N=27)	14	8	4	0	1

#### Comments on question III.1.1 - CASE tools

RTM, clearness, Various ERP packages

Requirements Management, Design, Code, Test, Costing

Rational Rose. Other miscellaneous tools like test coverage analyzers.

Case tools are used in the SDE for requirements, design, code and unit test as well as project management and metrics program.

Based on project size, platform, and client preferences, CASE tools would be used. This includes both third party and homegrown CASE tools

Rational Apex, ROSE, Teamwork, StP

Rational Rose, SLATE, CRADLE, plus home grown spin-offs of standard tools

Upper CASE tools like Rational Rose are used

Moving towards institutionalizing the use of case tools like Requisite Pro, Rational Rose, Test Tools and Code Coverage tools. However Configuration Management and Project Management tools have been standardized.

D2K, ROSE, S-Designer, ERWIN, SETS

Some use of Cadre

Various in-house tools and also from the market.

Rational Rose, Code generator

Cadre Teamwork RTSA, Ada; Rational Ada Tool set; Amadeus

Oracle Designer (aka Designer2000) used for application development projects

ERWin, System Architect, Rational Rose

Rational Suite

Rational Rose for design

Our Integrated Product Environment contains a selection of tools in order to improve productivity and quality as the program progresses.

Standard set of CASE tools for requirements, design, scm

Domain specific upper & lower tools are used. Quantitative baselines and other process assets are established for each domain and helps in effective planning for the use of these tools.

Rational ROSE, DOORS

Rational Rose; EasyCase

Teamwork and ObjectTeam, Rational.

### **Comments on question III.1.7 - Other process automation**

Configuration Mgt Tools, Test tools

Review process, CM process

Risk database, issue tracking system

Requirements traceability tools, Software CM tools

Estimation tool, Requirements tracing tool, static code analyzers, maintenance work request management tools are used

Process/Project database and time sheet automation is in place. In the process of finalizing the defect tracking system along with the metrics analysis tool.

Change control workflow management tool; Peer review workflow management tool; Process Definition tool; Process and Project Management tool

TTs : Web based time tracking and reporting system; RMS: web based resource management system; SPD : web based software process database and query system; PROMON : PB based process monitoring system modeled on the QMS, being currently ported to a web environment; SETS : automated code generation for PB, VB, Java, ASP; RRS : Remote Review System (pilot will be initiated to perform remote reviews); NETCARE: a web based system being developed to allow customers online view of our process metrics database

Audits/Actions databases, tied to CMM and ISO 9001

Process workflows.

Configuration Management tools, project planning, risk management and tracking tools, managing internal assessments, capturing and tracking process changes

Automated Test Tools

‘Review and Metrics Tool’ to capture and analyze defects during review/Inspection

Process non-compliance; QA DB; CM DB; Test DB; Test automation; Tasking; Resource Requests; Action Items; Personnal Suspenses; Correspondence Control Records; Fault Reporting System

Lotus Notes databases used with flavors of process enactment

MS project in work group mode, SCM tools such as ClearCase, code coverage tool

configuration management; costing; scheduling

domain specific development and test automation is in use where there is mature technology to support automation, e.g. protocol development & testing

In-house developed Web based process automation Tools

- ETS ( Effort Tracking System) to capture effort
- ARTS ( Automatic Review Tracking System) to capture Review process
- DTS (Defect Tracking System) to capture defect details during testing and post release

Regression/Test case automation

Training Information System, Skill Database and Resource Allocation

Metrics collection, reporting, and analysis. Including Orthogonal Defect Classification.

Collection of project metrics information.

Software Problem Report (SPR) System - consists of a main database that developers, customers and IV&V contractors share to coordinate problem reports - which is our primary mechanism for software changes. Both Product Assurance and the Software Team Lead keep independent databases of SPR related material for quality and consistency checks, and defect tracking/monitoring. All software build processes are fully automated via VAX DCL Scripts. A vast majority of design documentation is produced via automated tools (Fortran programs). An on-line database (MS Access) exists providing an index to all process material - which is key since 95% of our documentation is still paper.

**2. What, if any, tools or technologies that may prove to be important for performance excellence are currently being piloted in your organization?**

Rational Unified Process

OO lifecycle technologies. ClearQuest

Working on ONLINE SEARCHABLE DOCUMENT DATABASE of "Role Model" Documentation (Requirement Specification or Design or Project Plans or Proposals ) which is searchable on keywords of project types.

Speech recognition. Object-oriented design analysis. VDM - Validated Design through Modeling, by IFAD. Test Designer - Intusoft

Discover. Net Meeting

Knowledge management tools: HyperKnowledge, on-line bulletin boards of knowledge cells

process modeling

Metrics program mentioned above.

Inter group commitments tracking/workflow system, automation of process management, customer relationship management system

DOORS and Slate; ClearCase

Trade study tools; operation description tools

An integrated project and quality management tool; Metrics collection and reporting tool; Metrics repository; Defect Prevention Actions database and repository

Moving towards institutionalizing the use of case tools like Requisite Pro, Rational Rose, Test Tools and Code coverage tools. However Configuration management and Project Management tools have been standardized.

Process Automation Tools; Metrics Automation Tools

TECGEN : Test Case Generator; SET : Automated application generator; Test Data Generator; Requirements Management Tool; Regression Testing Tool

PDM tools

Tools for an integrated implementation of Requirements, Design, Code Generation and Testing related automation.

SPSS for quantitative analysis, CCC

Unit test tools - currently evaluating Ada Test by IPL Corp of the UK. 'Non Conformance Report' Tool for capturing and analyzing NCRs during quality audits. 'Meeting Minutes Tool' for capturing and analyzing action items. 'Status Reporting Tool' for sending project status to higher management. 'User Call Support Tool' for capturing and analyzing internal HW/SW complaints by the facilities group (IT Operations group).

Software Engineering Technical Repository; MS Exchange

System Network Administrator

Rational unified process

Problem Tracking tool

## JAVA

These are in common use.

- Protocol development & testing tools (vendor supplied and in-house developed)
- Rational tools (Rose, Clearquest, etc.)
- Project management workbench
- Internet technologies

1. Web based Defect Tracking System 2. SAP (ERP) Project Systems Module

Workflow Management, Knowledge Management

None - using established methods.

**3. Has your organization piloted or otherwise considered these or other tools or technologies but rejected them for common or standardized use? (Please briefly describe any such practices here, along with your reasons for rejecting them)**

Rational Unified Process (RUP). Ease of modification (difficult), using someone else's process not compatible to the way we do our business.

We have reviewed workflow and some integrated Case environments but they were rejected either due to cost benefit ratio or their cumbersome implementation or resource utilization.

Prior versions of above tools because they could not accommodate our typical programs (100+ software engineers). Many prior operating systems, computer systems, tools, and languages

Tool evaluation process has selected only a few but rejected many.

We evaluated COHESIONworkX from Digital several years ago. We attempted to implement and make use of a so-called integrated environment. It was a total failure. The tool worked under the light load of the pilot program, but when used in a production environment it constantly failed. The tool and the entire principle of an integrated automated environment was abandoned.

Third party tool on 'Requirements analysis' was evaluated and rejected as this contains lot of features that are not relevant to our use. Using simple 'Requirements Traceability Matrix' instead.

Scheduling tools, Process Automation Tools, Multimedia Technology, SONEX, Distributed Interactive Simulation, Voice Recognition Software, GUI Builder, Anti-virus software, Tiger Software Training Evaluator, (>60 other rejects in our DB)

Teamwork - it was standardized five years earlier and has now been eliminated from the standard since the technologies and projects we work on have different needs.

As a matter of standard practice, we are always evaluating new technologies - requirements management tools, CASE tools and such top the list. The most common reason for rejection comes down to the fact that it would be more work to migrate 21 years of program data to one of these systems that we'd ever get back in time savings. We've taken the time to evaluate the cost / benefit, but it never comes out in favor of introducing new technology. Most of our efforts are spent improving practices within the tools and methods currently available. A lot of it also has to do with the fact that we only have about 3-4 years left in the program.



## IV. Quantitative Analysis

1. How are the following quantitative analysis practices used in your software organization?

		Std	Comm	Not	D/K	N/A
1.1	Cost of quality analysis	11	11	14	1	0
1.2	Quality function deployment (QFD)	4	5	23	2	3
1.3	Orthogonal defect classification (ODC)	5	9	20	2	1
1.4	Other defect taxonomies (Please describe briefly here) (N=32)	15	6	5	3	3
1.5	Control charts (e.g., XbarR, XmR, u, Z, custom) (Please describe briefly here)	26	8	2	1	0
1.6	Other charting methods to understand “acceptable limits” of variation in predictability of performance (e.g., run or trend charts) (Please describe briefly here.) (N=36)	25	8	3	0	0
1.7	Prediction intervals	10	7	18	1	1
1.8	Confidence intervals	9	6	19	2	1
1.9	Designed experiments (Please describe briefly here)	1	4	26	5	1
1.10	Quasi-experimental methods (Please describe briefly here.)	0	2	25	6	4
1.11	Six sigma	7	4	21	5	0
1.12	Pareto analyses	20	11	6	0	0
1.13	Analyses of variance (e.g., ANOVA, ANCOVA, MANOVA)	7	7	19	3	1
1.14	Other multivariate methods (e.g., regression analyses, structural equations) (N=36)	2	9	20	3	2
1.15	Process modeling or simulation (e.g., system dynamics models, or spreadsheet based “what if” studies of process performance and impact analyses)	3	17	16	1	0

#### Comments on question IV.1.4 - Other defect taxonomies

Root Cause Based Analysis

Pareto Charts, Histograms, rework cost charts

Historical Company classification by defect severity and defect cause

in-house defined taxonomies

Inspection effectiveness, defect insertion as a function of activity and product type. Defect density over time.

We analyze defect insertion vs. defect detection, defect type by phase, cost to correct defects over the software life cycle

organization defined standard defect classification for software errors

Defect classification based on type, severity, cause category and phase of origin is used.

Severity

Defect models (e.g., STEER, SWEEP, DIRC)

severity levels

All STRs and Inspection Action Items require that the source of the defect be specified in a field called defect type and defect analysis. Data is tabulated and used with other data to determine common sources of problems

Defect Classifications as 'Major, Minor, Investigative' Various types of MTTR like MTTL (Mean Time To Localize' a bug)

Home-grown taxonomy

In-Process Quality as described above.

A procedure for defect analysis and prevention

defect classification

Comprehensive Hybrid of the ODC, IEEE and internal classification categories.

Defects are classified based on severity and on type of defects

#### **Comments on question IV.1.5 - Control charts**

XbarR

XbarR

Also Process Capability Charts

We are presently adding the XmR capability limits to our management review charts. In order to drive process improvement we have also included "take action" points that are tighter than the XmR capability limits.

Z and modified Z

U, Z, XmR

individual charts, XbarR, Z

We use control charts to assess variation over time for process stability and on several for the metrics that affect process stability.

XmR charts for in-process and acceptance defect density, schedule slippage, field errors, codewalkthru effort and defects

Details not available at this time.

We establish a mean and acceptable limits and study the variations periodically and at project milestones.

Statistical process control techniques are used on various process parameters.

Project Management

Bar, Z, C etc.

X, XmR, Z

SPC charts, using Excel

XmR, u-charts

Where we use control charts, XmR are used exclusively. Attempts to use multiple methods were too confusing. We can not expect every director and VP to be an expert in SPC. We only use control charts where they are appropriate. They do not tell the whole story. Reports need to be simple. We also make extensive use of comparative statistics such as Pareto.

Custom Control Chart with the control limits arrived based on the dispersion value (Standard Deviation)

Ubar

XbarR, XmR

XmR

Xbar

used for inspection data analysis

XmR control chart for schedule and effort analysis; U Control chart for review and test defect analysis

Control charts are standardized in testing and schedule analysis and becoming standard in earlier phases.

U charts or Z charts for Design Reviews, Code Reviews and Unit Testing

**Comments on question IV.1.6 - Other charting methods to understand “acceptable limits” of variation in predictability of performance**

Run Charts, Scatter Diagrams

Trend Charts

run charts with trends; workload activity charts with trends (used in our level-of-effort workloads). These charts show the number of tasks received, number closed, and number remaining open for each month. We also track work stoppage problems.

scatter diagrams, run charts, trend charts

Trend charts and multi-plots are used to track both programmatic metrics and process metrics in order to give advance warning of a potential problem using alert zones and automatic notification if a out of bounds condition occurs.

run chart, trend chart

Run charts, trend charts, Pareto charts, etc.

run and trend charts for cost, schedule, defects and productivity

Run charts

Trend Charts

Time line, Size analysis

S-Curve

SW Reliability Growth Model (latent defect projection)

trend charts

Run charts, Pareto, Trend analysis - anything that helps tell the story

Trend Charts are being used to predict future values.

trend charts, frequency charts, scatter diagrams

trend charts

Run and Trend Charts

bar charts, trend charts

run and trend charts: defects, staffing, CPI/SPI

used for productivity, quality analysis.

Histograms, Trend charts and scatter diagrams

Internal audits

Gompertz Curve, Fitment to Rayleigh Curve, Trouble Counter Measure Charts

Zone analysis.

We use trend charts with upper and lower bounds set that define 'management reaction' levels.

#### **Comments on question IV.1.9 - Designed experiments**

DOE or Design of Experiments

Commonly used for TCM purposes

6 sigma pilot covers this (or will when we get to that phase)

Not typically used but we have done some Six Sigma projects to do some causal analysis.  
This has been quite effective.

Used in Technology Change Management Pilots.

#### **Comments on question IV.1.10 - Quasi-experimental methods**

Prediction of learning curves

- 2. What, if any, quantitative analysis practices that may prove to be important for performance excellence are currently being piloted in your organization?**

Gompertz Curves for assessing delivered quality

CSIRO software metrics assessment

All described above form part of our Six Sigma Tool Kit

We're still learning what we can from the use of the XmR charts

Monte Carlo Risk Analysis

For 6 sigma, we will be considering: designed experiment, analyses of variance, simulation, and confidence intervals.

Six sigma, u charts

Currently piloting a more formalized method to calculate and predict total effort associated with software rework (include integration and test, changing of affected documents, etc.)

Formal Statistical Process Control for software management

Effort overrun, schedule overrun, and defect density practices were piloted and implemented org wide based on corporate quality norms. Review and testing efficiency is being piloted.

#### Orthogonal Defect Classification

Statistical model being piloted for certain projects to estimate workload. Benchmarking against industry data has been initiated

#### New integrated Project Management Tool

In-Process Quality as it relates to Defect detection, tracking and management. President's Goal Book relies on quantitative measurements of all business related quality goals.

#### Defect containment

Automated project data analysis which provides real-time, online information on the quantitative performance of projects. These are reviewed within the project teams for appropriate action. Sr. management reviews the data/actions planned at periodic intervals.

#### Estimation Tool based on past projects data

#### Design of Experiments using Orthogonal Arrays

### **3. Has your organization piloted or otherwise considered these or other quantitative analysis practices but rejected them for common or standardized use?**

We tried X(bar)-R charts in our automatic test equipment groups. The charts work excellent when applied to the ATE test results but were not beneficial for the purposes of managing our process.

SLIM Modeling tool - does not suit environment or application set.

Piloted and rejected use of McCabe complexity metrics due to inconsistency of tools, incompatible customer requirements, and questionable value-added.

We have used a standard set of project control metrics for over 10 years. These are essentially planned vs. actual metrics for each project. In 1996 when we first started looking at Level 4 requirements, we used an extensive GQM approach to select quantitative metrics best suited to our needs. This study showed that in many cases the most useful representation of data were Run Charts with trend analysis and Pareto comparisons. Only in a few cases did we select SPC Control Charts as providing the best approach to answering the questions that came out of GQM. Our consultants told us later that if we wanted to be at Level 4 we would need more control charts. As a result, we added additional control charts, but since we only have a

few software projects, and they only deliver to the customer about once every 3 to 5 years, there appears to be little gained from these charts. A requirement for Level 4 should not be the use of Control Charts. It should be to establish goals and use quantitative data to measure progress toward the goals.

The six sigma approach was used within the company (adopted from Motorola). It encouraged the use of arbitrary stretch goals without sufficient understanding of process capability and it encouraged sub-optimization of processes

Earlier (2 years back) we were doing analysis based on average or mean values, now as we are getting more samples of data, analysis focuses more into frequency domain of the data collected.

None. Right about the time we implemented a robust Quantitative Management Program was about the time we finished a major rehost effort. Since that rehost in 1996, we've had a total of 10 defects, and 14 other enhancements. Three defects a year simply does not provide enough data points to justify sophisticated analysis techniques. The same is true for other data points we'd monitor - they just don't occur at a frequency that justifies more than run time or trend charts.

**4. In which life cycle stages, if any, are control charts used in your organization?**

	<b>Stage</b>	<b>Control Chart</b>	<b>Std</b>	<b>Comm</b>	<b>Not</b>	<b>D/K</b>	<b>N/A</b>
4.1	Requirements	XbarR charts	8	1	28	0	0
		XmR charts	7	3	27	0	0
		u-charts	3	1	32	0	1
		Z-charts	3	1	31	0	1
4.2	Design	XbarR charts	8	2	27	0	0
		XmR charts	8	3	26	0	0
		u-charts	6	1	29	0	0
		Z-charts	6	0	30	0	1
4.3	Code	XbarR charts	8	3	26	0	0
		XmR charts	9	5	23	0	0
		u-charts	5	3	28	0	1
		Z-charts	6	0	30	0	1
4.4	Integration	XbarR charts	6	3	28	0	0
		XmR charts	7	2	28	0	0
		u-charts	4	1	31	0	1
		Z-charts	5	0	31	0	1

	<b>Stage</b>	<b>Control Chart</b>	<b>Std</b>	<b>Comm</b>	<b>Not</b>	<b>D/K</b>	<b>N/A</b>
4.5	Test	XbarR charts	6	4	27	0	0
		XmR charts	8	4	25	0	0
		u-charts	5	1	30	0	1
		Z-charts	6	0	30	0	1
4.6	Operations	XbarR charts	5	0	31	0	1
		XmR charts	7	1	29	0	0
		u-charts	1	0	35	0	1
		Z-charts	2	0	34	0	1

**5. What kinds of quality and performance measures are used in your software organization?**

		<b>Std</b>	<b>Comm</b>	<b>Not</b>	<b>D/K</b>	<b>N/A</b>
5.1	Cost performance index	29	2	6	0	0
5.2	Schedule performance index	33	1	3	0	0
5.3	Requirements stability (e.g., number of customer change requests or clarifications)	31	5	1	0	0
5.4	Process stability (e.g., number of changes or waivers from defined development processes)	21	8	8	0	0
5.5	Rework	16	11	10	0	0
5.6	Customer or user satisfaction	22	8	6	0	1
5.7	Staff morale	6	10	20	0	1
5.8	Defect density measures (e.g., from inspections and reviews, test results, other trouble reports, or field defect reports) (Please describe briefly here) (N=36)	35	1	0	0	0
5.9	Other quality or performance measures (e.g., MTTF, maintainability, interoperability, portability, usability, reliability, complexity, reusability, product performance, durability) (Please describe briefly here) (N=34)	18	8	7	0	1

#### **Comments on question IV.5.8 - Defect density measures**

Containment effectiveness

Composite (Weighted) Defect Indices based on severity; Applicable to all reviews, inspections & Testing. Also applicable to Post-Release Defects

We identify the phase injected and the phase defected for each defect. We identify whether the defect was found internally or externally. We also track the cost and schedule impacts of the defect correction activities. These activities help enable us to focus our SQM/DP activities.

Inspections of Design and Code and trouble reports from inspections and reviews, test results, field reports.

We calculate inspection effectiveness from inspections and calculate leakage as a percentage. We also calculate defect density as a function of time to watch trends.

Defect density as a XmR chart at organization level, both for in-process and acceptance phase defects for development projects, field errors for maintenance projects

Defects/KSLOC from reviews through delivery to the customer (our business is primarily development of new, fully-integrated systems. We are not typically involved in O&M activities).

Defects/KSLOC; Inspections/KSLOC; Latent (post-delivery) Defects/KSLOC

Delivered defect density; Phase-wide defect density

Defect per size(KSLOC /FP), Defects per phase - found in process, review, escaped from process, review

Review Effectiveness; Weighted Defects per Function Point; Weighted Defects per KLOC; Weighted Defects per Person Month

typical defects/ESS (equivalent source statements) throughout the life cycle

Various defect measures used with our in-house tools called DARTS.

from system testing, acceptance testing and post installation

Defects per KSLOC and Defect Per Requirement implemented are maintained using both trend and Control Charts for all inspection and test activities. Control charts monitor each

inspection and establish the norm for defects detected. Leakage is also monitored with control and trend charts by examining the defect origin activity vs. the defect found activity, or the percentage of defects leaking two or more activities.

Data collected during review, Inspection and testing against the size of the project.

/LOC in code inspections, integration test, and formal test, /test procedures, /rqmt errors span all detection phases and are tracked for life of system and beyond (errors/KSLOC)

Defect Rate tracked at each phase of the life cycle. Planned defects are documented using the SWEEP tool.

inspections, test results, trouble reports

Analysis on parameters and Metrics like Defect removal efficiency, Defect Density, Post release defects

Integration and system test faults and field faults

Reviews, testing and customer reported

Defects per 1000 lines of code by product (e.g., requirements, design, code)

Defects discovered are a part of the DRB minutes for a given code inspection. If it was an incorrect implementation of an existing problem (e.g. already an SPR), it's handled at the DRB level. If it's a new defect, a new SPR is generated and tracked as a defect.

#### **Comments on question IV.5.9 - Other quality or performance measures**

Reliability, Complexity, Functionality, MTTR (popular), MTBF (not so common, but started), Interoperability, Maintainability

Reliability and complexity are used in proposal cost and schedule estimation.

These measures are used at a system level -- software contributes to them.

We calculate complexity to determine that the module will be maintainable and understandable.

review effectiveness (early detection, backlog management index, change request responsiveness index

Do all of the above, with some variation due to individual customer preferences.

## Technical Performance Measures (TPMs)

Cumulative defect removal efficiency; Phase-wise defect removal efficiency; MTTF for product development

MTTF,....

Memory size, product performance

MTTF, reliability, etc.

Also Technical Performance Measures (Throughput, capacity, etc.)

We have initiated measuring reliability and its related performance measures like failure intensity, MTTF

Complexity by unit, using a control chart is a measure of maintainability that we use.

Reusability

complexity and size stability

MTBF, # failures, and other custom reliability measures

Complexity, product performance, reusability.

complexity, product performance

product quality goals that are critical to a project are decided at the planning phase and formally tracked on a monthly basis.

Complexity, reliability & plot performance

reliability, interoperability, usability , availability, serviceability, optimizability

Defect containment, Quality indicator/analysis

We used to keep careful track of program size and performance, but that's been so stable over the last 4 years it's not actively monitored now. In the 80's, code efficiency (in terms of compiled size) and throughput were most important, since this was an embedded system with limited memory (64K). That took precedence over maintainability and complexity - it didn't matter. What mattered was reliability, size and speed.

## V. Other Practices

### 1. What other important practices, if any, does your organization follow on a common or standardized basis?

Training planning and conduct, as well as evaluation of effectiveness of training

We conduct a Process and Technology Review for each proposal and after contract award to insure the organizational standard process is being implemented and that the standard toolset is being implemented.

Intranet covering all TCS sites/branches is being built. As a result, knowledge bases will get integrated.

Proposing and bidding new business; Process improvement program; Standardized program review by executive management (including process deployment); Enterprise metrics (up to company level management)

Defect Prevention Process and supporting tool are standardized.

- Monthly review of Quality aspects by senior management in every business unit.
- Monthly process assessment in each project to identify strengths and improvement opportunities.
- Defect Prevention action teams in every business unit to leverage defect prevention across projects.
- Focus on building new competencies via Internal Competency Center.
- Focus on building domain expertise via Global Competency Center.
- Annual Project Managers conference to share experiences, best practices and other information.
- Intranet for information sharing.

Customer Feedback on completion of a project, Customers meet in a common forum once a year, internal audit conducted at customer locations for our staff, HR Climate survey once in 2 years, Annual cultural activities to boost the staff morale.

Process modeling, process assets reuse, process automation tools, metrics automation tools, on-line process asset library, process consultant for each project, Organizational interoperability - process definition for all support activities (finance, HR etc.) to reduce cycle time - Support organizations work with anticipation and co-ordination

QES : Phil Crosby's Quality Education System; NIP : National Induction Program; PQI : Personal Quality Initiative; LC's : Leadership Circles; Bootcamp : Rigorous 6-8 week training program; QMS for TM : Quality Management System training for team members

1. Learnings (what went wrong, what went well) and project assets are systematically gleaned from projects, categorized for easy access and made available to all project managers who are facilitated during Project Initiation and Look-ahead meetings to avoid/assimilate practices.
2. Tools developed for internal use in projects are evaluated for propagation or enhancement in the organization by forming focus groups.
3. The GQM methodology is used to set meaningful goals and use appropriate measures to track them for different types of projects.
  - Very detailed and simply explained metrics guidelines are made available to all practitioners. Performance metrics are widely communicated and best practices recognized. Metrics experts are available for consultation. Statistical analysis is done on defects and schedule commitments to study trends and common causes of variation.
4. Optimal Automation and access to all to process engineering activities via
  - process change request and its management
  - process release management
  - process document access and navigation
5. Traceability is maintained for every process change with its incorporation in standard processes and release for use. It is valued based on a defined mechanism and awards given for Best Change suggestion.
6. High focus on Customer Satisfaction achieved through
  - Periodic Customer Satisfaction Survey, analysis of results and process improvements to enhance services
  - Analysis of unsolicited customer feedback for process improvements
7. Business Effectiveness Surveys are conducted to ascertain employee satisfaction /morale and to take appropriate actions for improvement.
8. Quality group conducts SQA effectiveness surveys on QA group interaction with projects, technology usage surveys and process effectiveness surveys to get feedback from practitioners. Learnings database is continuously enhanced. New process assets are created for reuse.
9. One of the major strengths of the organization is the Individual Skill Development Program which is created by individual in consultation with their managers and tracked at the organization level along with training, supported by extensive global training opportunities.
10. All non-conformances and weaknesses are studied for the root causes and specific instances are taken up for organization level defect prevention by training, asset creation or management focus.
11. Defect prevention training on root cause analysis, look ahead meetings and planning guidelines on triggers etc are provided.

12. Technology changes are proactively taken up and processes for communication, piloting, roll outs, feedback, transitioning and sharing of experiences are well defined and supported by full time resources. Up-to-date technology for day to day activities is available to all practitioners. Access to worldwide intra net provides vast information on all topics in addition to access to Internet.
13. Organization-wide involvement and commitment to quality is visible in terms of management participation, resources availability and rewards and a focus on all aspects of internal and external customer satisfaction.

Knowledge sharing through various fora (Quarterly SQA meet, Tech talk, Testing consortium, etc)

risk management; defect causal analysis; project estimation; requirements coordination meeting; formal tasking, requests for resources, requests for tools, and requests to evaluate new technology

comprehensive root cause analysis for every identified software defect

Continuous Process Improvements, Quality-Page articles, Technical information sharing

SE-CMM

Technology planning processes; Internal bi-annual process assessments; people management practices; rewards & recognition systems for meeting/exceeding organizational goals

We conduct detailed data based project performance analysis. This is being facilitated by a role called Quality improvement facilitator. During this meeting entire project team participates. This practice has helped to a great extent on driving the importance of accurate data collection, data analysis and to plan action based on the data analysis. Another important practice is effective use of intranet for providing information on Quality system and process assets

Increasing Process Buy-in by disseminating info on the success of statistical predictive capability on a continuous basis; Very strong and structured Stage Reviews

Statistical correlation of customer satisfaction scores to internal performance metrics.

Most of our practices translate directly to the CMM - the one difference is that we take several practices to an extreme just because of the domain we're in. Code Inspections (DRB), Defect Prevention, Testing, CM and QA are the cornerstones of our process.

2. **Would you like to share any further comments about the definition, use, or improvement of the software processes in your organization? (Please describe fully.)**

We share our process with the customer by making the command media available to them online and having them participate in our process improvement activities including SEPGs.

We are leading the standardization effort for software assessment training, processes, and follow-up across Raytheon Systems Company (RSC). We are key participants in the development and deployment of high-maturity processes and training across RSC.

We have an online Quality System, which includes the following:

- Quality System documentation
- Project Knowledge Base (consisting of organization baselines and goals, project data, defect prevention repository, best practices, sample documents and tools information)
- general quality information
- role-based Quality System Training Kits
- Suggestion Box via which any member of Satyam can send their quality related suggestions to SEPG.

There is no organizational process as generally understood. There is a process modeling methodology that uses the process assets library to produce a unique process for a product

The QMS is a web based system, and is accessible to all employees. Organization wide participation in the SPI program is encouraged through training. The SPI program is managed thru a web based system, right from logging suggestions, review of suggestions, logging the disposition and rationale of disposition, the action plan, and the final outcome. It is accessible to all employees and is completely transparent, though maintained by SEPG. It supports a very flexible reporting system.

Process improvement is an extensive focus with two models of ISO 9001 and CMM being implemented hand in hand with the whole organization involved

Average improvements per process in last 2 years = 4.2

Planning to connect all departments and locations with the intranet. Going for an enterprise wide web based work-flow automation system.

With the rapid pace at which technology is changing, domain specific processes and technology changes have to be quickly incorporated with the organizations standard processes and require dedicated resources and focus. Dedicated technology and software

1. Use of Process automation tools like ETS and ARTS 2. Bottom up approach of Process definition 3. Contribution from task forces to address 'pain areas' 4. Efficient and effective use of Organizational Intranet 5. Structured training programs

## VI. People Issues

### 1. How many people are employed in your organization?

**Total number of employees (N=34)**

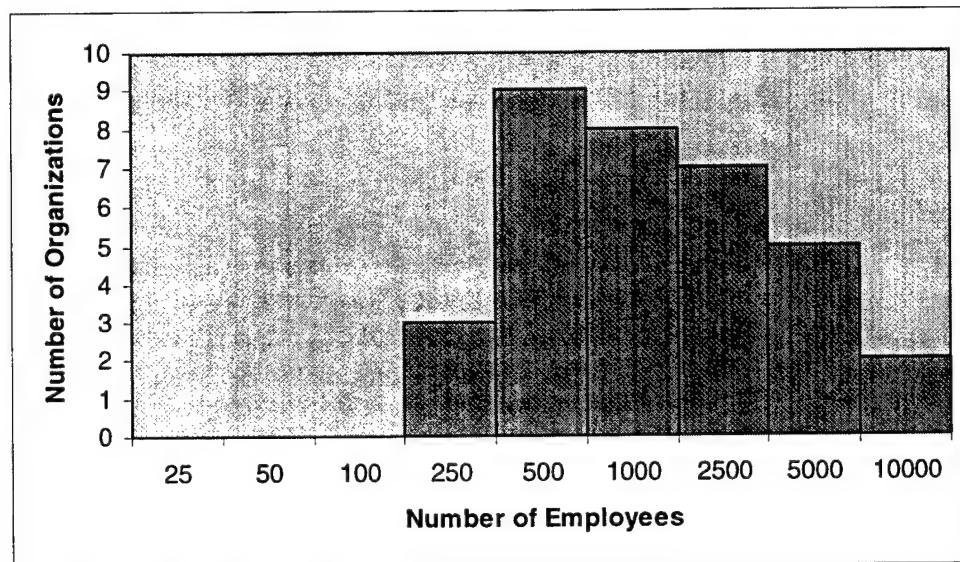
Mean 1601

Median 950

Max 7500

Min 140

StDev 1727



*Figure 1: Questions VI.1 - Total Number of Employees*

**Number primarily engaged in software development, maintenance, or support (N=35)**

Mean 810

Median 400

Max 5000

Min 17

StDev 1075

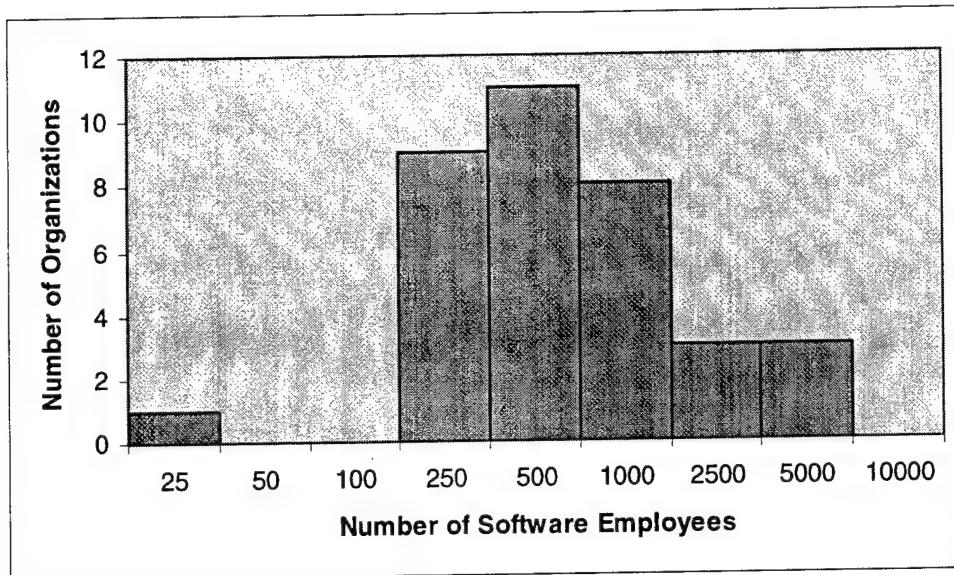


Figure 2: Questions VI.1.2 - Number of Software Employees

**Total full-time-equivalent software employees (full timers plus the hours worked by part timers & consultants) (N=35)**

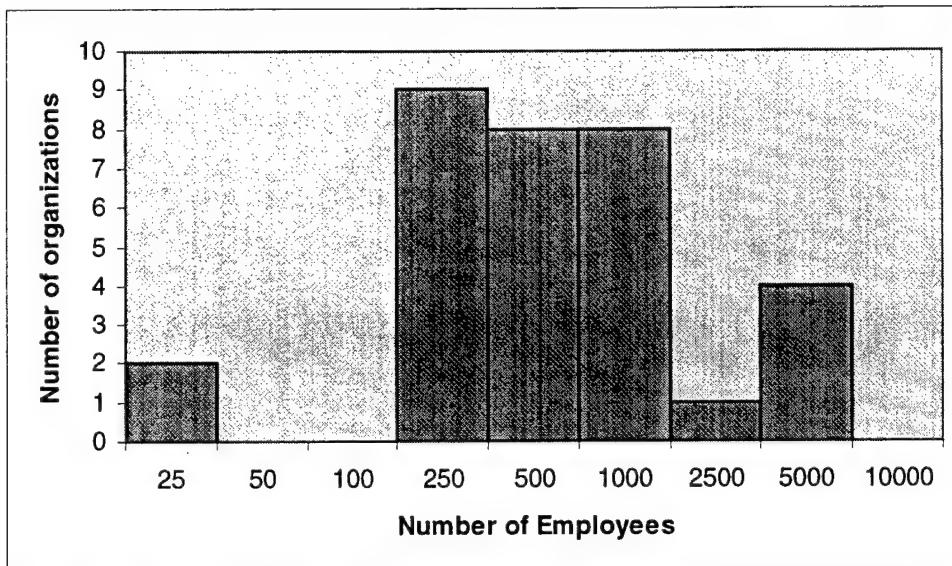
Mean 772

Median 355

Max 4000

Min 1

StDev 1043



*Figure 3: Question VI.1.3 - Total Number of Full-Time-Equivalent Software Employees*

2. **Approximately how many days of induction training does your organization provide to new hires? (N=34)**

Mean 17

Median 6

Max 90

Min 1

StDev 24

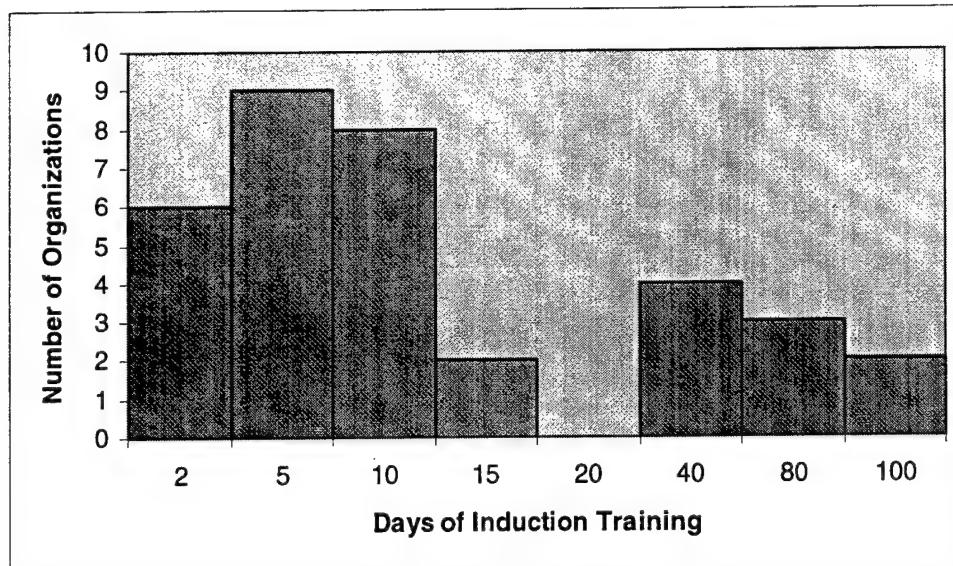


Figure 4: Question VI.2 - Days of Induction Training

3. Does your organization have a formal mentoring program for new hires? (e.g., long term relationships with experienced and knowledgeable mentors) (N=35)

Yes = 17      No = 18

**Comments on question VI.3 – Formal mentoring program**

Buddy system

Starts off with Lunch with your "buddy" who becomes a mentor. Mentor is usually a lead designer of the same program but not a manager. Mentoring is mostly built on informal relationships, has specific targets to achieve but its effectiveness is not formal

Multiple formal mentoring methods including mentoring by: company president, program managers, specialist and architects.

currently setting up a program

Technical mentoring is provided by the project lead. Other organization related mentoring is provided by appointed mentors

75% of new hires are assigned a mentor by their manager.

Two month training program for fresh recruits on various technology areas, process, and soft skills.

The Project leader imparts domain knowledge for new recruits as identified in the project plan

Just looking at formalizing this (informal now)

The IBM Mentoring Program consists of following steps:

- Employee works with Manager to determine if mentoring is the best way to meet personal and business needs
- Mentor and Protege are matched
- The Mentor and Protege develop a detailed Mentoring Action Plan that leads to the accomplishment of the agreed-upon objectives
- Then they perform according to the plan and monitor progress
- Proteges update their personal development information (for example, the skills database, development plan) to reflect newly obtained skills, knowledge, and experience. Mentors update their personnel development information (e.g. leadership skills) as appropriate

Following are being practiced:

- Identifying a buddy on a new employee joining the company.
- 'On the Job training coordinators' on technical coaching.

Following is being piloted:

- Formal trained mentors for all employees.
- Training on mentoring

Requires trained mentors to be trained in mentor process; accountability of mentor in performance plans; documented objectives of training sessions; tracking of completed mentor training; satisfaction surveys for mentors and mentees; minimum one year relationship for new hires; no-fault separation clause; defined mentor selection criteria.

Currently being improved, now mentor must self-nominate by filling out compliance form and then signed off by management.

The program is not 'formal' in the sense no one tracks the status formally. However, it is in common practice. Each new hire after the induction & domain training, is assigned a sr. team member who will induct him/her into the practices in the organization.

3 months

We have structured training programs. We have systematic process for inducting a person to Software engineering methods as well as his for his role in project

Mentoring Program with defined guidelines and feedback.

They are usually mentored by the Team Lead for their group, or another expert in their field. It is expected and planned for that experienced employees will spend time with new hires. Our orientation program also facilitates this, since each major functional area is reviewed.

**4. Approximately how many days of continuing education do employees get per year in your organization? (N=32)**

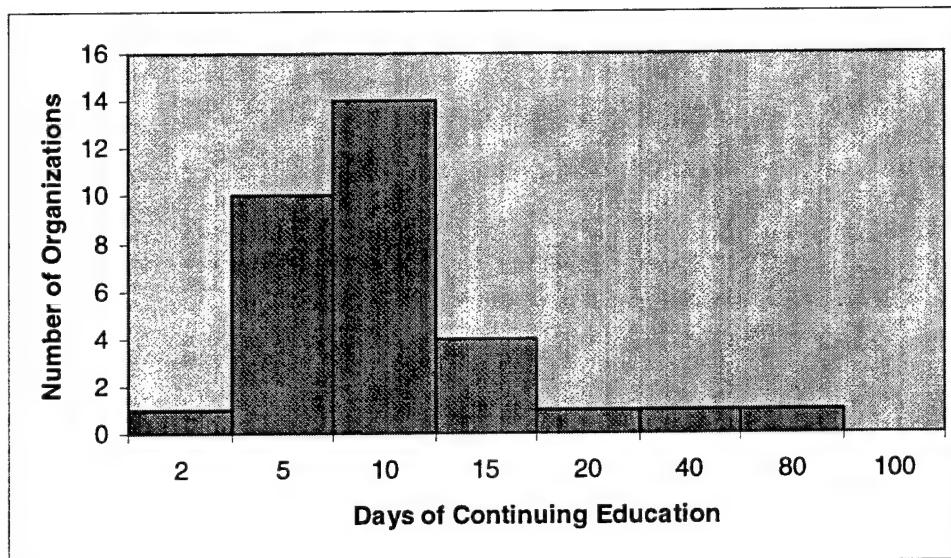
Mean 11

Median 10

Max 80

Min 2

StDev 13



*Figure 5: Question VI.4 - Days of Continuing Education*

**5. Does your organization provide its employees with required training in ...? (N=35)**

	<b>Yes</b>	<b>No</b>
Technical skills of software engineering	35	0
Management skills	33	2

	Yes	No
Domain knowledge	28	7
Interpersonal skills	28	7
Principled negotiation	19	16
Team building	26	9

6. Approximately what proportion of the total software staff in your organization left the organization during the past 12 months? (i.e., what was your yearly turnover or attrition rate?) (Percent) (N=33)

Mean 13

Median 14

Max 40

Min 1

StDev 8

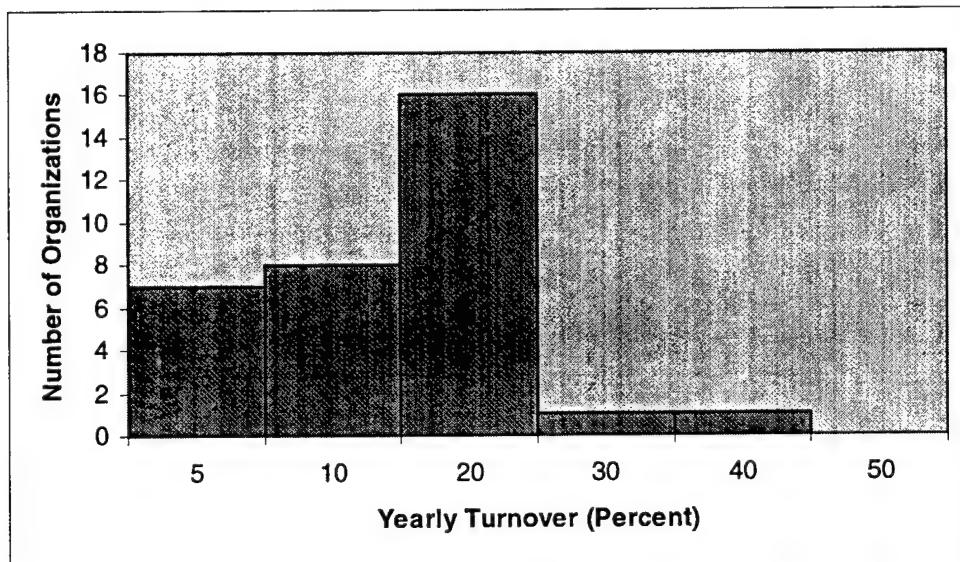


Figure 6: Question VI.6 - Yearly Turnover (Attrition)

7. Approximately what proportion of the total software staff in your organization joined the organization during the past 12 months? (i.e., what was your yearly growth rate?) (Percent) (N=32)

Mean 25

Median 20

Max 100

Min 0

StDev 22

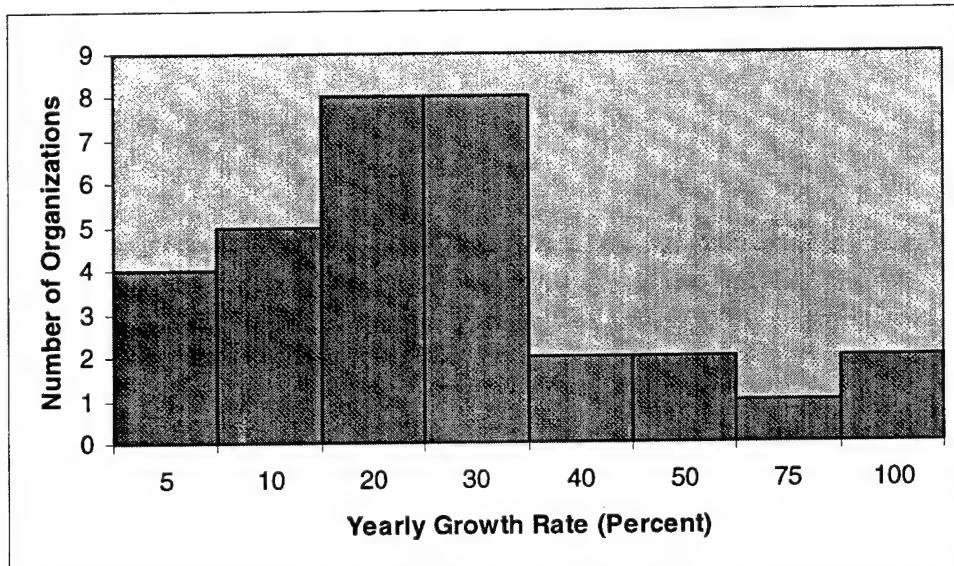


Figure 7: Question VI.7 - Yearly Growth Rate

8. In what kinds of work spaces do the technical staff in your organization typically work? (Percent)

	... in private offices	... in shared offices	... in individual cubicles	... in other open work spaces
N	35	35	35	35
Mean	10	18	51	19
Median	5	0	40	0
Max	36	87	100	100
Min	0	0	0	0
StDev	10	31	40	36

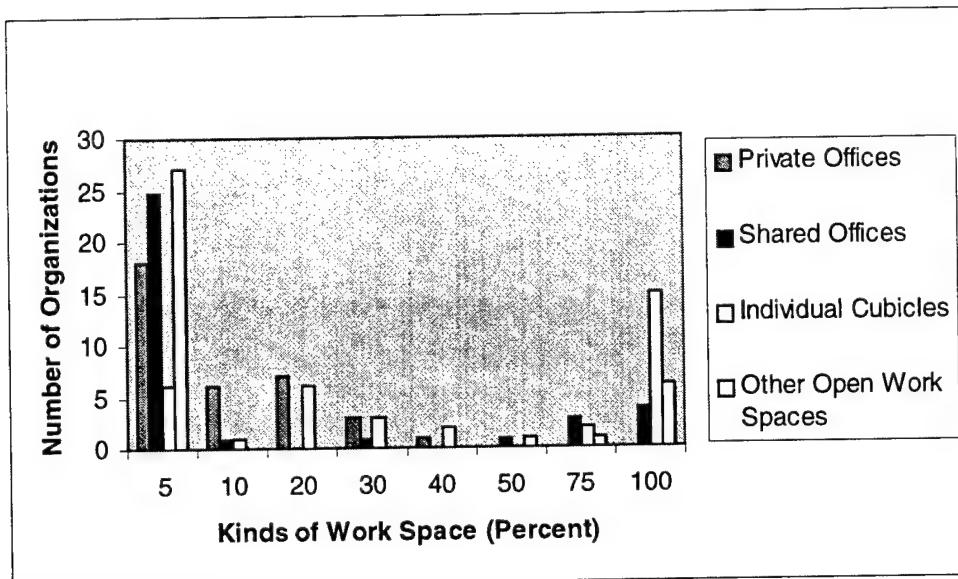


Figure 8: Question VI.8 - Kinds of Work Space

9. About how many people typically work on the projects in your organization?

	... on large projects	... on average size projects	... on small projects
N	31	31	31
Mean	283	197	88
Median	80	50	10
Max	3500	2500	1500
Min	10	8	1
StDev	671	487	270

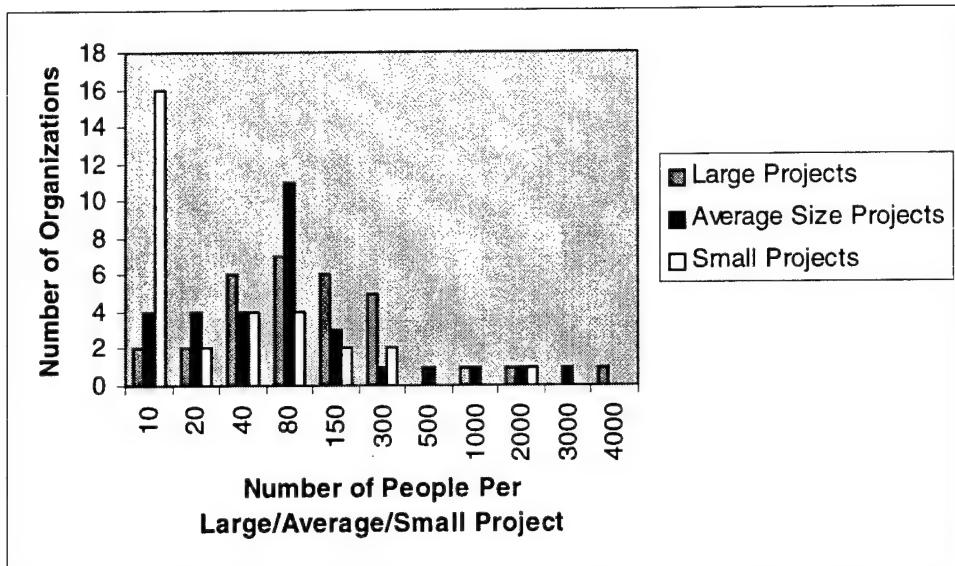


Figure 9: Question VI.9 - Number of People Per Project

10. About how long do projects typically last in your organization? (Months)

	... for large projects	... for average size projects	... for small projects
N	32	32	31
Mean	39	18	8
Median	24	12	4
Max	240	67	44
Min	10	3	1
StDev	44	16	10

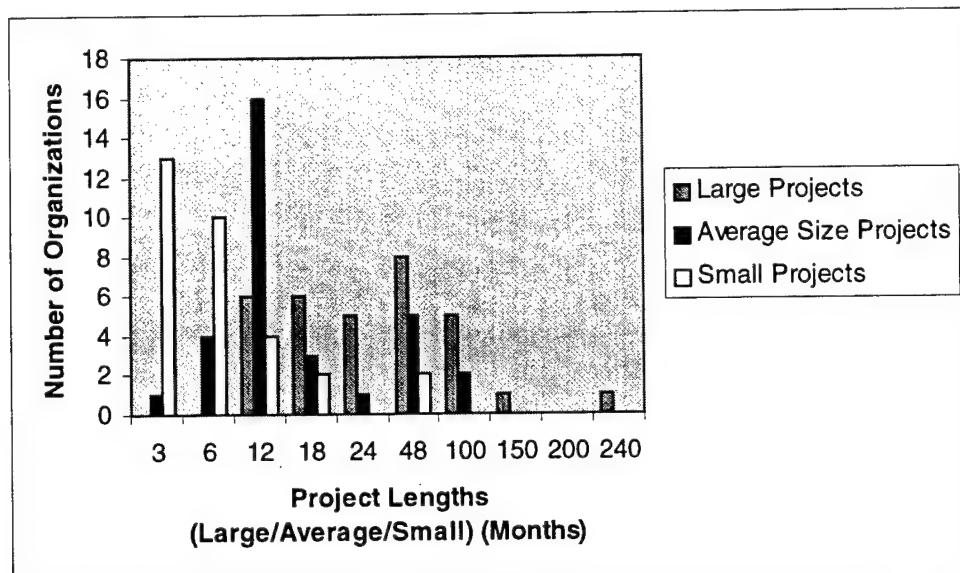


Figure 10: Question VI.10 - Project Lengths

## VII. Background Information

1. Approximately how much of your organization's business is devoted to software (or the software in software intensive systems)? (Percent)

	Development	Maintenance	Acquisition
N	33	30	14
Mean	62	28	15
Median	70	23	8
Max	100	90	80
Min	5	2	3
StDev	27	20	21

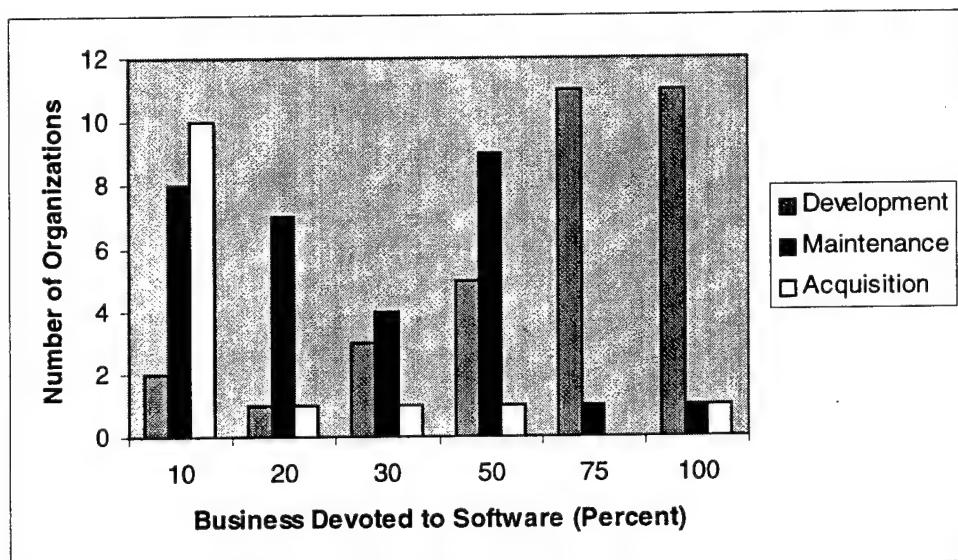


Figure 11: Question VII.1 - Percent of Business Devoted to Software

2. Approximately how long ago did your organization begin work on improving its software processes? (Months) (N=34)

Mean	103
Median	79
Max	360
Min	24
StDev	78

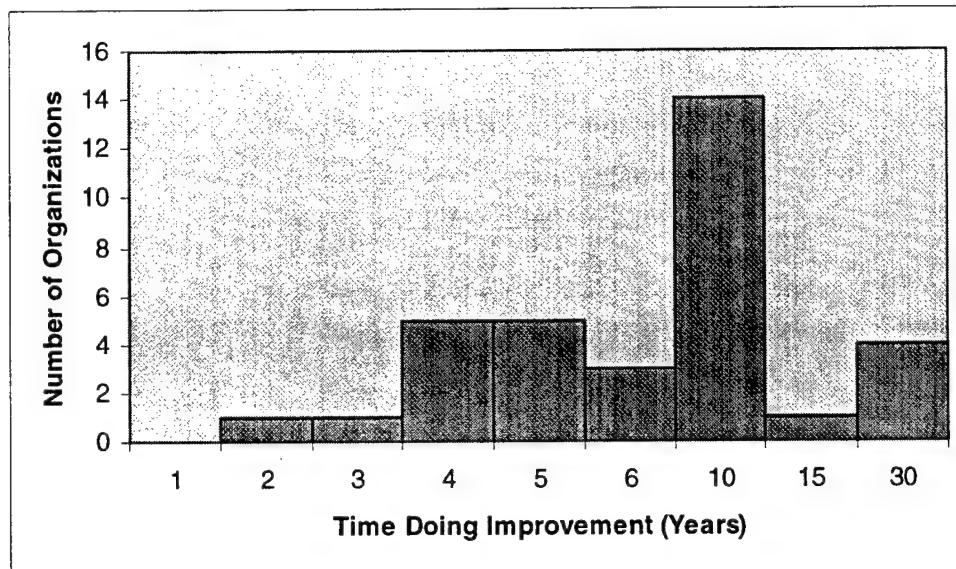


Figure 12: Question VII.12 - Length of Time Doing Process Improvement

3. In what year was your organization assessed at ... ? (N=35)

	≤1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
Maturity level 1	4	2	2							
Maturity Level 2	1	1	2	3	1	1				
Maturity level 3	1		1	1	2	4	5	2		
Maturity Level 4						2	1	6	8	6
Maturity Level 5	1						1	2	2	9

Transitions between levels reported	N
1-2-3-4	4
1-2-4	1
1-3-4	2
1-3-5	1

Transitions between levels reported	N
2-3-4	2
2-3-5	2
3-4-5	2
3-5	4
4 <sup>5</sup>	10
4-5	2
5	5

4. How is your organization best described? (N=35)

Defense contractor	10
Other government contractor	2
Department of Defense or military organization	3
Other government agency	0
Commercial shrink-wrap	1
Custom software development	9
"In-house" or proprietary development or maintenance	1
Other industry or commercial (e.g., manufacturing; health or pharmaceutical; finance, insurance, or real estate; wholesale or retail trade)	5
Other (Please describe briefly)	4

**Comments on question VII.4 – Organization description**

Medical Scanners

---

<sup>5</sup> It is possible that those respondents reporting only the Level 4 or 5 achievement had earlier assessments performed but failed to report the dates of those assessments. It is also possible that the organization waited until reasonably certain of achieving a high maturity rating before performing a "formal" assessment such as a CBA IPI.

Software for Banking and Financial industry

Communications software

Telecommunications

Wireless communications

Develop software for overseas clients from India

24% defense; 33% civil; 11% commercial; 32% international

Software development and maintenance services

We are a subsidiary of Motorola and a development group for the various units of Motorola which need software services.

International lab on hire

Software Services in System software, Telecom, Datacom and Embedded Technologies.

Funding has been about 80% Air Force, 20% NASA

**5. For what major application domains does your organization develop, maintain, or acquire software or software intensive systems? (N=35)**

Management Information Systems (e.g., systems supporting business operations such as payroll, accounts receivable, payable, inventory, or logistics)	12
Real Time Applications (e.g., process control, manufacturing, automation, guidance systems for avionics or radar)	23
Embedded Systems (e.g., software running in consumer electronic devices, vehicles, fuel control, military systems)	23
Other (Please describe briefly)	6

**Comments on question VII.5 – Application domains**

Medical Scanners

System Software (OS, Compilers..)

Basic banking functions

System software like operating system, network, protocols, data communication

Hardware design

Telecom, datacom, networking

Software Services in System software, Telecom, Datacom and Embedded Technologies.

Large scale telecommunications software development

**6. Does the organization concentrate its efforts on ... ? (N=35)**

A core product line or application domain (e.g., switches, guidance systems, information systems, or database systems)	25
A core technology (e.g., distributed systems, real time embedded systems, object oriented design, or simulators)	19
Life or mission critical systems	13
Extremely large or complex systems	19
New or poorly understood domains or technology	13
Other special focus (Please describe briefly)	4

**Comments on question VII.6 – Organizational concentration**

Electronic Design Automation

Smart card Systems

Telematics

Communications

Medical Scanners And Related Software Development from device drivers to real time systems to Windows based GUIs to Network Packages to Web Applications (Wide Technology Domain)

Offshore Development Centre for client

Reengineering and Maintenance

To provide services in the area of System software, Telecom, Datacom and Embedded Technology domains.

Selected Core technology areas

**7. How is the organization structured? (N=35)**

Functional (i.e., by common specialties such as finance or engineering)	15
Product (i.e., by units responsible for a product or product line)	18
Customer group (e.g., targeting customers such as the US Navy or General Motors)	12
Territorial (e.g., Northeastern marketing zone)	6
Matrix (i.e., a mixed project and functional organization)	20
Process (i.e., by flow of work such as IPPD)	3
Other (Please describe briefly)	2

**Comments on question VII.7 – Organizational structure**

Within a product area, organized by IPTs

By specific Technology areas

**8. Does your organization have a total quality management (TQM) or other similar program? (N=34)**

For the assessed organization	7
At a higher corporate or similar parent level	6
Both of the above	16
Neither	5

**9. Is the organization ISO 9001 certified (Quality Management Systems)? (N=35)**

Yes = 27      No = 8

**In what year was the organization first certified?**

1989	1	1993		1997	6
1990		1994	6	1998	3

1991		1995	5	1999	2
1992		1996	4		

**10. What, if any, other quality or process improvement models, approaches, or emerging standards does your organization use in its improvement efforts? (N=35)**

ISO/IEC 12207 "Software Life Cycle Processes" (or IEEE)	14
ISO/IEC 15504 "Software Process Assessment"	5
ISO/IEC 15288 "System Life Cycle Processes"	3
ISO/IEC 15939 "Software Measurement Processes"	2
SEI SE-CMM	11
INCOSE SE-CAM	3
EIA SE-CM	8
FAA iCMM	0
Software Acquisition CMM	4
People CMM	10
Other (Please describe briefly)	5

**Comments on question VII.10 – Other improvement models/standards**

CMM-I, EIA-632

SW-CMM

CCQMS : Crosby's Complete Quality Management System

TickIT

Planning to evaluate PSP and P-CMM

ISO 9001

We are planning for new initiative to implement PSP and TSP models

TL 9000

Malcolm Baldrige equivalent (Tata Business Excellence Model)

<b>REPORT DOCUMENTATION PAGE</b>			<i>Form Approved OMB No. 0704-0188</i>
<p>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</p>			
1. AGENCY USE ONLY (LEAVE BLANK)	2. REPORT DATE February 2000	3. REPORT TYPE AND DATES COVERED Final	
4. TITLE AND SUBTITLE The 1999 Survey of High Maturity Organizations		5. FUNDING NUMBERS C — F19628-95-C-0003	
6. AUTHOR(S) Mark C. Paultk, Dennis Goldson, David M. White			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213		8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-2000-SR-002	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116		10. SPONSORING/MONITORING AGENCY REPORT NUMBER N/A	
11. SUPPLEMENTARY NOTES			
12.A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS		12.B DISTRIBUTION CODE	
<p>13. ABSTRACT (MAXIMUM 200 WORDS)</p> <p>Over the last few years the Software Engineering Institute has investigated the high maturity practices of Maturity Level 4 and 5 software organizations via assessments, site visits, workshops, and surveys. This report summarizes the observations from the 1999 survey of high maturity organizations. Areas covered in the survey include management, engineering, tools and technology, quantitative analysis, and people issues. A specific area of interest is statistical process control, which is addressed in some detail in this report, but the observations cover a variety of engineering and management practices, including issues outside the scope of the Capability Maturity Model for Software.</p>			
14. SUBJECT TERMS High Maturity, Software Organizations Management Practices, Engineering Practices, Software CMM		15. NUMBER OF PAGES 84	
16. PRICE CODE			
7. SECURITY CLASSIFICATION OF REPORT  UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE  UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT  UNCLASSIFIED	20. LIMITATION OF ABSTRACT  UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)  
Prescribed by ANSI Std. Z39-18  
298-102